AI-driven Mentorship and Support System for Girls in Rural India Preparing for IIT JEE

Submitted in partial fulfillment of the requirements of the degree of

BACHELOR OF ENGINEERING
In
COMPUTER ENGINEERING

By

Komalika Acharya

Jay Aslaliya

Vedant Deshmukh

Yash Vaskar

Supervisor:

Ms. SAKSHI SURVE
(Assistant Professor, Department of Computer Engineering, TSEC)

Computer Engineering Department
Thadomal Shahani Engineering College
University of Mumbai
AY 2023-2024

AI-driven Mentorship and Support System for Girls in Rural India Preparing for IIT JEE

Submitted in partial fulfillment of the requirementsof the degree of

BACHELOR OF ENGINEERING
In
COMPUTER ENGINEERING

By

Group No : 35

2003002	**Komalika Acharya**
2003012	**Jay Aslaliya**
2003044	**Vedant Deshmukh**
2003185	**Yash Vaskar**

Supervisor:

MS. SAKSHI SURVE
(Assistant Professor, Department of Computer Engineering, TSEC)

THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

Computer Engineering Department
Thadomal Shahani Engineering College
University of Mumbai

AY 2023-2024

CERTIFICATE

This is to certify that the project entitled **"AI-driven Mentorship and Support System for Girls in Rural India Preparing for IIT JEE"** is a bonafide work of :

2003002	**Komalika Acharya**
2003012	**Jay Aslaliya**
2003044	**Vedant Deshmukh**
2003185	**Yash Vaskar**

Submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"BACHELOR OF ENGINEERING"** in **"COMPUTER ENGINEERING"**.

(Ms. Sakshi Surve)
Supervisor

(Prof. Tanuja Sarode)
Head of Department

(Prof. G.T. Thampi)
Principal

Project Report Approval for B.E

Project report entitled ***AI-driven Mentorship and Support System for Girls in Rural India Preparing for IIT JEE*** by

2003002	**Komalika Acharya**
2003012	**Jay Aslaliya**
2003044	**Vedant Deshmukh**
2003185	**Yash Vaskar**

is approved for the degree of ***"BACHELOR OF ENGINEERING" in "COMPUTER ENGINEERING"***.

Examiners

1._____

2._____

Date:

Place:

Declaration

I declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. we also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

1) _____
(Signature)
Komalika Acharya – 2003002

2) _____
(Signature)
Jay Aslaliya – 2003012

3) _____
(Signature)
Vedant Deshmukh - 2003044

4) _____
(Signature)
Yash Vaskar – 2003185

Date:

Abstract

The traditional way of preparing for IIT JEE examination is through classes, tuitions and one on one learning but its not the same for girls in rural areas. They are not allowed to study and if they are allowed, girls don't have resources and monetary support to help them prepare for the examination. So, our project is trying to solve this problem by providing the girls in rural area who will help them prepare for examination.

In the pursuit of equality in STEM education, this project introduces an AI-driven Mentorship and Support System designed exclusively for young girls in rural areas with ambitions to conquer the rigorous IIT JEE examination. Recognizing the multifaceted challenges they face, this platform is meticulously crafted to offer tailored mentorship, personalized guidance, motivational reinforcement, and a wealth of educational resources. By addressing the unique hurdles confronted by these aspirants, the system aims to empower them to surpass their circumstances, boost their confidence, and elevate their performance in this highly competitive endeavor. This initiative represents a pivotal step towards bridging the gender gap in STEM education and fostering the next generation of female leaders in science and technology.

List of Figures

Chapter 1

Introduction

1.1 Introduction

The Joint Entrance Examination (JEE) is a highly competitive entrance examination in India for admission to various undergraduate engineering and architecture programs at prestigious Indian Institutes of Technology (IITs), National Institutes of Technology (NITs), and other top engineering institutions across the country. There are two main types of JEE exams:

JEE Main: JEE Main is the first stage of the examination. It serves as a screening test for admission to various undergraduate engineering programs.
JEE Advanced: JEE Advanced is the second stage of the examination and is the pathway to secure a seat in one of the IITs

JEE exams are known for their rigor and competitiveness, and they are the gateway to some of the most prestigious engineering institutions in India. Candidates often prepare for these exams rigorously, and coaching institutes offer specialized coaching for JEE aspirants. It's important to check the official websites and notifications for the most up-to-date information and any changes in the examination pattern or eligibility criteria.

Girls in rural areas face a myriad of challenges when it comes to accessing education. Limited access to schools, transportation issues, and concerns about personal safety can often deter girls from pursuing their education. Outdated gender stereotypes and traditional societal norms can lead families to prioritize boys' education over girls', and early marriages and pregnancies further hinder girls' educational opportunities. The lack of female teachers, language and cultural barriers, and economic constraints can compound the problem. Addressing these challenges requires a holistic approach, involving investment in rural infrastructure, raising awareness about the value of girls' education, providing scholarships, and creating an inclusive and safe learning environment for all. Ensuring access to quality education for girls in rural areas is crucial for their empowerment and the overall development of these communities.

1.2 Aims & Objective

The system is not just a passive educational platform; it's a dynamic, comprehensive ecosystem with a holistic approach that seeks to revolutionize the educational landscape for girls facing unique challenges. At its core, it's about empowerment. It's about giving these girls the tools, resources, and personalized guidance that they need to not only break down barriers but to shatter glass ceilings. The educational resources provided aren't just generic materials; they're meticulously curated to align with the diverse needs, backgrounds, and learning styles of these girls.

Moreover, the system extends its reach beyond mere academics. It becomes a mentor, a confidant, and a source of unwavering motivation. Through personalized guidance, it helps these young women navigate the intricate pathways of science and technology, providing insights, encouragement, and mentorship.

By boosting their confidence, it isn't just raising their academic performance; it's also empowering them to embrace leadership roles, fostering innovation, and ensuring diversity in the tech world. This system is, in essence, a catalyst for change, a beacon of hope that symbolizes the transformative power of education when it is truly personalized and inclusive. Its vision is to see these girls not only thriving in the realm of science and technology but becoming trailblazers and role models, inspiring future generations to chase their dreams without hesitation.

1.3 Scope

Implement an AI-driven recommendation system that assesses individual learning styles, strengths, and weaknesses, and generates personalized learning paths for each Girls(student).
Establish a mentorship network that connects girls with experienced women in STEM professions, offering guidance, support, and career advice.
Conduct ongoing research to assess the impact of the project, gathering data on academic performance, confidence levels, and career aspirations of the girls involved.

Chapter 2

Review of Literature

2.1 Domain Explanation

1. Education and Examination Landscape:

Several online platforms like Khan Academy, Coursera, and edX offer educational content in a wide range of subjects. They use AI to provide personalized learning experiences. However, they may not be tailored specifically to the needs of IIT JEE aspirants in *rural areas*.

2. Rural Education Challenges:

Students in rural areas face a plethora of challenges in accessing quality education. These include limited access to well-equipped schools with qualified teachers, long commutes that consume valuable study time, economic constraints that hinder educational expenses, and a dearth of essential educational resources like libraries and laboratories.

Furthermore, rural schools often grapple with teacher shortages and language barriers that affect the quality of instruction. In addition to these educational challenges, infrastructure-related issues such as the lack of electricity and proper classrooms, unreliable internet connectivity, and shortages of teaching aids and educational materials further compound the difficulties faced by rural students. These inequalities in educational opportunities perpetuate an urban-rural educational divide, highlighting the pressing need to address these issues to ensure equitable access to quality education and empower rural students to reach their full potential.

3. AI and Machine Learning:

AI and Machine Learning (ML) have brought transformative changes to the realm of education, particularly in the realm of personalized learning. AI-driven technologies have enabled the customization of learning experiences in various ways. They analyze individual students performance and preferences, leading to tailored content delivery, adaptive assessments, and virtual tutor support. These technologies also automate grading, predict student outcomes, enhance collaboration, and support data-driven decision-making in educational institutions.

The impact of AI in education is palpable through platforms like Khan Academy, Coursera, and Duolingo, which have demonstrated their effectiveness in improving learning outcomes, expanding

access to education, and making learning engaging and interactive. These platforms exemplify the power of AI and ML in education, which continues to evolve and shape the future of teaching and learning.

2.2 Review of Existing Systems

There are many coaching institution in India that offers JEE courses online. For many students, more than anything, personal attention and mentoring matters the most and few available platforms fulfils that need. It provides personalized attention through counselling, student performance monitoring among others.

Khan Academy: [6]Khan Academy offers a wide range of educational content and exercises, making it accessible to a broad audience. However, it lacks the specialization necessary for competitive exams like IIT JEE and may not address the unique needs of rural girls preparing for this particular examination.

Coursera and edX: [7]These platforms provide access to a variety of courses from universities worldwide. While they offer high-quality content, they do not specifically target IIT JEE preparation and may not offer the personalized support and guidance needed for such a competitive exam.

Competitive Exam Prep Apps: Apps like BYJU'S and Toppr cater to exam preparation, but they may not address the specific challenges of rural girls. Their content is generalized and caters to a general audience.

Government Initiatives: Government-led e-learning platforms, such as India's "Diksha," focus on general education but do not have any mentors and do not offer the personalized support and guidance needed for such a competitive exam.

2.3 Limitations of Existing Systems

Our proposed system stands out from platforms like Khan Academy due to its specific focus on girls studying in rural areas with aspirations to crack the highly competitive IIT JEE examination. While existing platforms provide valuable resources, our system offers a tailored approach that addresses the unique challenges and needs of this demographic, including gender-related barriers and limited access to quality education and mentoring.

What sets us apart is our deep personalization, AI algorithms that analyze individual preferences and recommend suitable mentors, offline capabilities for low-connectivity regions, continuous adaptation through feedback loops, and an explicit commitment to gender inclusivity and empowerment. This project not only provides academic support but also offers crucial motivational guidance and customized learning materials designed for success in the IIT JEE exam.

1. Khan Academy :

Positive Reviews: Khan Academy is praised for its extensive video library and easy-to-understand explanations. The platform is known for its flexibility, allowing learners to study at their own pace.
Limitations: Khan Academy may not offer specialized content for competitive exams like the IIT JEE, and it may lack localized resources. Additionally, it does not provide the level of personalized mentorship and support needed for aspirants in rural areas.

2. Coursera :

Positive Reviews: Coursera receives acclaim for its wide range of courses offered by top institutions. Learners appreciate the opportunity to gain knowledge from renowned professors and experts.
Limitations: Coursera courses may not be tailored specifically for the IIT JEE examination, and their academic depth might not align with the exam's requirements. Additionally, accessibility in low-connectivity regions could be a challenge.

3. BYJU'S[8]:

Positive Reviews: BYJU'S is favored for its engaging video lessons and interactive content, making it appealing to a broad range of students.
Limitations: The platform may not cater to the specific demands of competitive exams like the IIT JEE. It might lack the specialized content and mentorship needed for aspirants in rural areas.

4. DreamBox[9]:

Positive Reviews: DreamBox excels in providing adaptive math programs for K-8 students, adapting to their responses and offering customized math education.

Limitations: Its primary focus on math education may limit its applicability for the comprehensive preparation required for the IIT JEE examination.

5. Smart Sparrow[10]:

Positive Reviews: Smart Sparrow is known for its adaptive e-learning platforms, particularly in the STEM field. It personalizes learning experiences to promote engagement and understanding.

Limitations: While effective in STEM education, Smart Sparrow's specialization may not address the diverse subject matter of the IIT JEE examination. Additionally, access in low-connectivity areas may pose challenges.

6. Physicswallah[11]:

Positive Reviews: Physicswallah is highly regarded for its physics tutorials and explanations. The platform is known for its comprehensive coverage of physics topics, making it a valuable resource for students preparing for competitive exams.

Limitations: While Physicswallah excels in education and content, it does not provide personalized mentorship.

Chapter 3

Proposed System

3.1 Analysis/Framework

The proposed AI-driven mentorship and support system for girls studying in rural areas with aspirations to crack the IIT JEE examination is a commendable initiative. However, like any project, it may face several potential limitations and challenges:

1. Technological Infrastructure
2. Digital Literacy
3. Internet Connectivity
4. Tailoring to Local Syllabus
5. Quality of Offline Materials
6. Access to Mentors
7. Socioeconomic Barriers
8. Data Privacy and Security
9. Language Barriers
10. Long-term Sustainability

The proposed project's AI-driven mentorship system has the potential to be a transformative force in empowering girls in rural areas to pursue their aspirations. By offering personalized guidance, motivational support, and tailored resources, it can address specific needs and challenges. The use of advanced AI algorithms for individual analysis and adaptive learning is a forward-thinking approach. The project's offline capabilities are a crucial feature, as they bridge the digital divide, ensuring access for girls in low-connectivity areas. Additionally, the integration of continuous feedback loops allows for ongoing improvement, making the system more effective over time.

However, the project needs to carefully consider and address the potential limitations mentioned. Ensuring technological access and digital literacy, maintaining high-quality offline materials, and addressing socioeconomic and cultural barriers are critical.

Data privacy, cybersecurity, and long-term sustainability should also be central concerns.

In summary, the project's success will depend on its ability to navigate and overcome these challenges while leveraging its strengths. By doing so, it can make a significant positive impact on the educational and aspirational journey of girls in rural areas preparing for the IIT JEE examination.

3.2 Design Details

1. External Entities:

- Student: This is the user group seeking mentors.
- Mentors: The mentors available in the system.
- Admin: The system administrator responsible for managing the platform.

2. Processes:

- Registration and Authentication:
  - Students and mentors can register and log in to the system. This process includes user authentication.
- Recommendation Engine:
  - This is the core component that recommends mentors to students based on their profiles and preferences.
- Feedback and Ratings:
  - After a mentorship interaction, students can provide feedback and ratings for mentors.
- Admin Management:
  - The admin can manage user accounts, handle disputes, and monitor the overall system.

3. Data Stores:

- User Profiles Database:
  - Stores information about students and mentors, such as their profiles and preferences.
- Interaction History Database:
  - Keeps records of past interactions and feedback/ratings given by students.
- Recommendation Algorithm Data:
  - Stores data used by the recommendation engine to make personalized mentor recommendations.

4. Data Flows:

- Students provide their profile information during registration and updates.
- Mentors provide their profile information during registration and updates.
- The recommendation engine uses student and mentor profiles to suggest mentor-student matches.

- Students can leave feedback and ratings, which are stored in the Interaction History Database.

- Admins can access user data and monitor system activities.

- The recommendation engine may periodically update its recommendation algorithm based on user interactions.

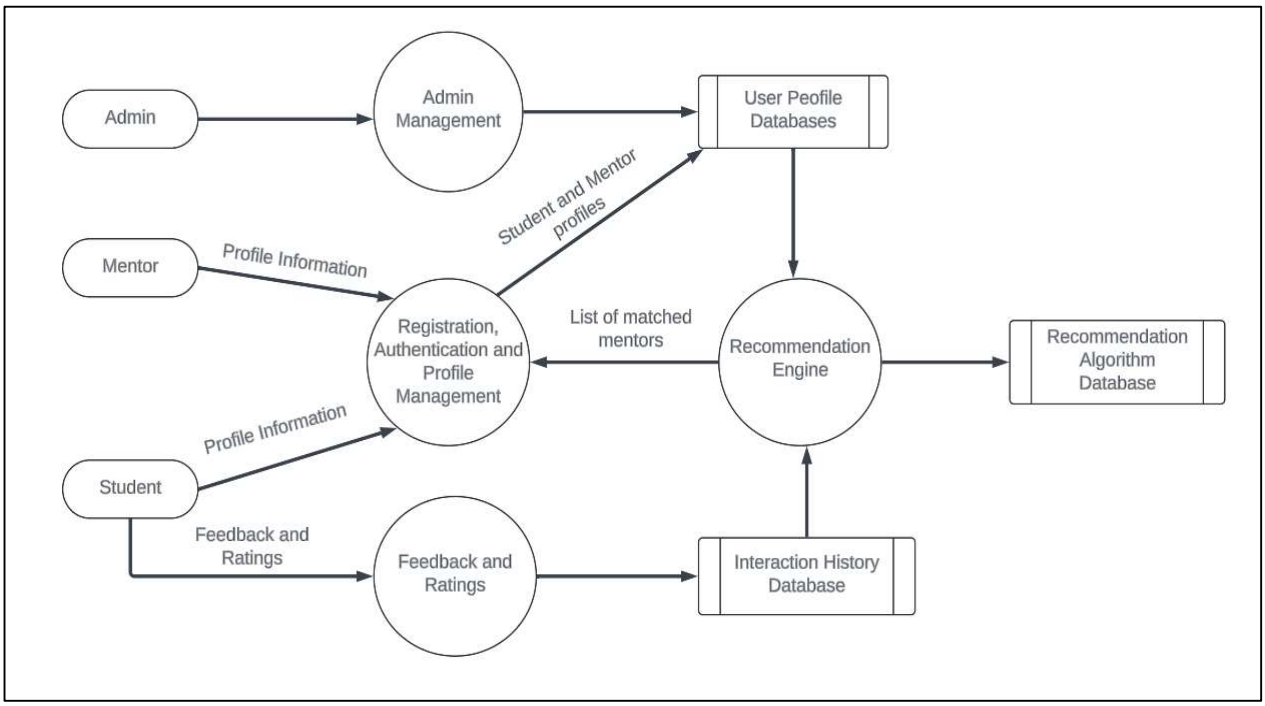

Figure 3.1 DFD Diagram

3.3 Methodology

1. Needs Assessment and User Profiling : Begin with a comprehensive needs assessment by collaborating with educational experts and engaging with the target audience. Understand the unique challenges and requirements of girls in rural areas preparing for the IIT JEE exam. Collect data on their existing knowledge levels, learning preferences, and technological access.

2. Content Development and Syllabus Alignment : Develop high-quality content tailored to the IIT JEE syllabus. Collaborate with subject matter experts to ensure content aligns with the exam's requirements. Create a curriculum that covers all relevant topics comprehensively.

3. AI System Development : Design and develop the AI system that will power the mentorship and support platform. Utilize advanced machine learning algorithms to analyze individual strengths,

weaknesses, and learning preferences. Create an adaptive learning engine that can recommend personalized study plans, learning modules, and practice materials.

4. Offline Access Integration : Develop and test the offline mode of the platform, ensuring that girls in low-connectivity regions can access learning materials and receive mentorship support even without a continuous internet connection. This may involve the development of mobile apps and offline resources.

5. User Interface and Experience Design : Create an intuitive and user-friendly interface for the platform, keeping in mind that some users may have limited digital literacy. Ensure that the platform is visually engaging and easy to navigate.

6. Mentorship Program Implementation : Establish the mentorship program, recruiting experienced educators, subject matter experts, and successful exam-takers to provide guidance, support, and doubt resolution to the girls. Develop a system for pairing mentors with mentees based on compatibility and subject needs.

7. Data Privacy and Security Measures : Implement robust data privacy and security measures to protect user information and ensure compliance with data privacy regulations. Clearly communicate the system's privacy policy to users.

8. User Training and Support : Provide training and support resources to ensure that girls are comfortable using the platform. Offer user guides and access to customer support for technical issues.

9. Cultural Sensitivity and Inclusivity : Ensure that the platform's content and approach are culturally sensitive and respectful of local norms and values, recognizing the diversity of rural regions.

This comprehensive methodology outlines the steps needed to create an effective AI-driven mentorship and support system for girls in rural areas preparing for the IIT JEE examination. By carefully implementing each step and continuously improving the platform, the project can have a lasting, positive impact on the educational journey of these girls.

Chapter 4

Implementation Details

4.1 Experimental Setup

4.1.1 Dataset Description/Database Details

The dataset created in this project report is a comprehensive collection of data pertaining to educators who provide instruction for the IIT JEE (Indian Institutes of Technology Joint Entrance Examination) entrance exam. The dataset comprises information on multiple teachers and their respective attributes, allowing for a detailed analysis of the teaching workforce associated with IIT JEE preparation.

We created the dataset on our own by writing a python script which will create a dataset like a real time dataset. The reason for doing so is that readymade dataset was not available anywhere possible.

It was done mainly by random function in python. A list of all possible realistic values of a feature was created and then a random function was used to populate the dataset row.

The dimensions of dataset are – 20 (rows) x 1000 (columns)

```
import os
import csv
import random

Name = ["Komalika Acharya", "Jay Aslaliya", "Vedant Deshmukh", "Yash Vaskar", "Sakshi Surve", "Tasneem Mirza", "Ujwala Bharambhe", "Anil
Chhangani", "Aejaz Khan", "Arti Deshpande", "Darakshan Khan", "Arti Deshpande", "Tanuja Sarode", "Khalid Ansari", "Ram sir", "Seema
Koulkar", "Pullarabinda Mahapatra", "Ayush Dodia", "Krishana Dave", "Prajwal Dhule", "Piush Paul", "Prem  Acharya", "Kartik Jolapara",
"Ayush Singh","Hinata Shoyo", "Tobio Kageyama", "Gojo Satoru", "Geto Sugur", "Itadori Kyuji", "Megumi Fushiguro", "Tanjiro Kamado", "Son
Goku", "Prince Vegeta", "Piccolo Daimao", "Son Gohan", "Trunks Vegeta", "Majin Buu", "Lord Beerus", "Kale Caulifla", "Saiyan Cabba", "Black
Frieza", "Green Broly", "Angel Whis", "Grand Priest", "Grand Zenos", "Master Roshi", "Krillin", "Manjiro Sano", "Ken Ryuguji", "Takemichi
Hanagaki", "Tanjiro Kamado", "Nezuko Kamado", "Zenitsu Agatsuma", "Inosuke Hashibira","Giyu Tomioka", "Shinobu Kocho", "Kyojuro Rengoku",
"Tengen Uzui", "Mitsuri Kanroji", "Muichiro Tokito", "Gyomei Himejima","Obanai Iguro", "Sanemi Shinazugawa", "Muzan Kibutsuji", "Kokushibo",
Preferred_Days = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]

Preferred_Days2 = ["Monday, Tuesday", "Monday, Wednesday", "Monday, Thursday", "Monday, Friday", "Monday, Saturday", "Monday, Sunday",
"Tuesday, Monday", "Tuesday, Wednesday", "Tuesday, Thursday", "Tuesday, Friday", "Tuesday, Saturday", "Tuesday, Sunday", "Wednesday,
Monday", "Wednesday, Tuesday", "Wednesday, Thursday", "Wednesday, Friday", "Wednesday, Saturday", "Wednesday, Sunday", "Thursday, Monday",
"Thursday, Tuesday", "Thursday, Wednesday", "Thursday, Friday", "Thursday, Saturday", "Thursday, Sunday", "Friday, Monday", "Friday,
Tuesday", "Friday, Wednesday", "Friday, Thursday", "Friday, Saturday", "Friday, Sunday", "Saturday, Monday", "Saturday, Tuesday", "Saturday,
Wednesday", "Saturday, Thursday", "Saturday, Friday", "Saturday, Sunday", "Sunday, Monday", "Sunday, Tuesday", "Sunday, Wednesday", "Sunday,
Thursday", "Sunday, Friday", "Sunday, Saturday"]

Preferred_Days3 = ["Monday, Tuesday, Wednesday", "Monday, Tuesday, Thursday", "Monday, Tuesday, Friday", "Monday, Tuesday, Saturday",
"Monday, Tuesday, Sunday", "Monday, Wednesday, Thursday", "Monday, Wednesday, Friday", "Monday, Wednesday, Saturday", "Monday, Wednesday,
Sunday", "Monday, Thursday, Friday", "Monday, Thursday, Saturday", "Monday, Thursday, Sunday", "Monday, Friday, Saturday", "Monday, Friday,
Sunday", "Monday, Saturday, Sunday", "Tuesday, Wednesday, Thursday", "Tuesday, Wednesday, Friday", "Tuesday, Wednesday, Saturday", "Tuesday,
Wednesday, Sunday", "Tuesday, Thursday, Friday", "Tuesday, Thursday, Saturday", "Tuesday, Thursday, Sunday", "Tuesday, Friday, Saturday",
"Tuesday, Friday, Sunday", "Tuesday, Saturday, Sunday", "Wednesday, Thursday, Friday", "Wednesday, Thursday, Saturday", "Wednesday,
Thursday, Sunday", "Wednesday, Friday, Saturday", "Wednesday, Friday, Sunday", "Wednesday, Saturday, Sunday", "Thursday, Friday, Saturday",
"Thursday, Friday, Sunday", "Thursday, Saturday, Sunday", "Friday, Saturday, Sunday"]

Preferred_Days4 = ["Monday, Tuesday, Wednesday, Thursday", "Monday, Tuesday, Wednesday, Friday", "Monday, Tuesday, Wednesday, Saturday",
"Monday, Tuesday, Wednesday, Sunday", "Monday, Tuesday, Thursday, Friday", "Monday, Tuesday, Thursday, Saturday", "Monday, Tuesday,
Thursday, Sunday", "Monday, Tuesday, Friday, Saturday", "Monday, Tuesday, Friday, Sunday", "Monday, Tuesday, Saturday, Sunday", "Monday,
Wednesday, Thursday, Friday", "Monday, Wednesday, Thursday, Saturday", "Monday, Wednesday, Thursday, Sunday", "Monday, Wednesday, Friday,
Saturday", "Monday, Wednesday, Friday, Sunday", "Monday, Wednesday, Saturday, Sunday", "Monday, Thursday, Friday, Saturday", "Monday,
Thursday, Friday, Sunday", "Monday, Thursday, Saturday, Sunday", "Monday, Friday, Saturday, Sunday", "Tuesday, Wednesday, Thursday, Friday",
"Tuesday, Wednesday, Thursday, Saturday", "Tuesday, Wednesday, Thursday, Sunday", "Tuesday, Wednesday, Friday, Saturday", "Tuesday,
Wednesday, Friday, Sunday", "Tuesday, Wednesday, Saturday, Sunday", "Tuesday, Thursday, Friday, Saturday", "Tuesday, Thursday, Friday,
Sunday", "Tuesday, Thursday, Saturday, Sunday", "Tuesday, Friday, Saturday, Sunday", "Wednesday, Thursday, Friday, Saturday", "Wednesday,
Thursday, Friday, Sunday", "Wednesday, Thursday, Saturday, Sunday", "Wednesday, Friday, Saturday, Sunday", "Thursday, Friday, Saturday,
Sunday"]

Preferred_Days5 = ["Monday, Tuesday, Wednesday, Thursday, Friday", "Monday, Tuesday, Wednesday, Thursday, Saturday", "Monday, Tuesday,
Wednesday, Thursday, Sunday", "Monday, Tuesday, Wednesday, Friday, Saturday", "Monday, Tuesday, Wednesday, Friday, Sunday", "Monday,
Tuesday, Wednesday, Saturday, Sunday", "Monday, Tuesday, Thursday, Friday, Saturday", "Monday, Tuesday, Thursday, Friday, Sunday", "Monday,
Tuesday, Thursday, Saturday, Sunday", "Monday, Tuesday, Friday, Saturday, Sunday", "Monday, Wednesday, Thursday, Friday, Saturday", "Monday,
Wednesday, Thursday, Friday, Sunday", "Monday, Wednesday, Thursday, Saturday, Sunday", "Monday, Wednesday, Friday, Saturday, Sunday",
"Monday, Thursday, Friday, Saturday, Sunday", "Tuesday, Wednesday, Thursday, Friday, Saturday", "Tuesday, Wednesday, Thursday, Friday,
Sunday", "Tuesday, Wednesday, Thursday, Saturday, Sunday", "Tuesday, Wednesday, Friday, Saturday, Sunday", "Tuesday, Thursday, Friday,

Preferred_Days6 = ["Monday, Tuesday, Wednesday, Thursday, Friday, Saturday", "Monday, Tuesday, Wednesday, Thursday, Friday, Sunday",
"Monday, Tuesday, Wednesday, Thursday, Saturday, Sunday", "Monday, Tuesday, Wednesday, Friday, Saturday, Sunday", "Monday, Tuesday,
Thursday, Friday, Saturday, Sunday", "Monday, Wednesday, Thursday, Friday, Saturday, Sunday", "Tuesday, Wednesday, Thursday, Friday,
Saturday, Sunday"]

Preferred_Days7 = ["Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday"]

Mode = ["Video lectures", "one-on-one", "doubt solving", "live lectures"]

Mode2 = ["Video lectures, one-on-one", "Video lectures, doubt solving", "Video lectures,live lectures ", "one-on-one, doubt solving",
"one-on-one, live lectures", "doubt solving, live lectures", "Video lectures, one-on-one, doubt solving", "Video lectures, one-on-one, live
lectures", "Video lectures, doubt solving, live lectures", "Video lectures, one-on-one, doubt solving, live lectures"]
```

Figure 4.1.1 Dataset Values

```
chunk=[]

with open(csvFilePath, 'w',newline='') as f_object:
    writer_object = csv.writer(f_object)
    writer_object.writerow(["Name", "Age", "Region", "Languages", "Languages-2", "Subject", "Preferred_Time", "Preferred_Time-2",
                            "Preferred_Days", "Preferred_Days-2", "Preferred_Days-3", "Preferred_Days-4", "Preferred_Days-5",
                            "Preferred_Days-6", "Preferred_Days-7", "Mode", "Mode-2", "ExperiencePhysics",
                            "ExperienceChemistry", "ExperienceMaths"])
    f_object.close()

for i in range(10000):

    age=(random.randint(23,70))

    l1=Languages[random.randint(0,(len(Languages)-1))]
    l2=Languages[random.randint(0,(len(Languages)-1))]
    while(l1==l2):
        l2=Languages[random.randint(0,(len(Languages)-1))]
    languages2=(l1+', '+l2)

    subject=Subject[random.randint(0,(len(Subject)-1))]
    result = [x.strip() for x in subject.split(',')]
    if("Physics" in result):
        experiencePhysics=(random.randint(0,47))
    else:
        experiencePhysics=0
    if("Chemistry" in result):
        experienceChemistry=(random.randint(0,47))
    else:
        experienceChemistry=0
    if("Maths" in result):
        experienceMaths=(random.randint(0,47))
    else:
        experienceMaths=0

    chunk.append([
        Name[random.randint(0,(len(Name)-1))],
        age,
        Region[random.randint(0,(len(Region)-1))],
        Languages[random.randint(0,(len(Languages)-1))],
        languages2,
        subject,
        Preferred_Time[random.randint(0,(len(Preferred_Time)-1))],
        Preferred_Time2[random.randint(0,(len(Preferred_Time2)-1))],
        Preferred_Days[random.randint(0,(len(Preferred_Days)-1))],
        Preferred_Days2[random.randint(0,(len(Preferred_Days2)-1))],
        Preferred_Days3[random.randint(0,(len(Preferred_Days3)-1))],
        Preferred_Days4[random.randint(0,(len(Preferred_Days4)-1))],
        Preferred_Days5[random.randint(0,(len(Preferred_Days5)-1))],
        Preferred_Days6[random.randint(0,(len(Preferred_Days6)-1))],
        Preferred_Days7[0],
        Mode[random.randint(0,(len(Mode)-1))],
        Mode2[random.randint(0,(len(Mode2)-1))],
        experiencePhysics,
        experienceChemistry,
        experienceMaths
    ])


with open(csvFilePath, 'a',newline='') as f_object:
    writer_object = csv.writer(f_object)
    writer_object.writerows(chunk)
    f_object.close()
```

Figure 4.1.2 Dataset Creation

The dataset created in this project report is a comprehensive collection of data pertaining to educators who provide instruction for the IIT JEE entrance exam. The dataset comprises information on multiple teachers and their respective attributes, allowing for a detailed analysis of the teaching workforce associated with IIT JEE preparation.

We created the dataset on our own by writing a python script which will create a dataset like a real time dataset. The reason for doing so is that readymade dataset was not available anywhere possible.

It was done mainly by random function in python. A list of all possible realistic values of a feature was created and then a random function was used to populate the dataset row.

The dimensions of dataset are – **20 (rows) x 1000 (columns)**

Key Features/Variables:

- **Name –** Full name of the mentor.
- **Age –** Age of the mentor.
- **Region –** Region from which mentor belongs.
- **Language –** Languages in spoken by mentor or is fluent in.
- **Subjects taught –** Subjects taught by mentor. Can be either Physics, Chemistry, or Math, or either two, or all three.
- **Preferred Time –** Preferred time for mentor to teach or guide which can be either Morning, Afternoon, or Evening, or either two, or all three.
- **Day for teaching –** Preferred Days for teaching. Can be any number of days in a week from 1 to 7.
- **Experience in Teaching –** Experience in teaching in each subject mentioned.
- **Mode –** Mode in which mentor is comfortable with to help student. Can be either doubt-solving, one-on-one, or live lectures, or either two, or all three.

	A	B	C	D	E	F	G	H	I	J	K
	Name	Age	Region	Languages	Languages-2	Subject	Preferred_Time	Preferred_Time-2	Preferred_Days	Preferred_Days-2	Preferred_Days-3
	Trunks Vegeta	58	Andaman and Nicobar Islands	Assamese	Santali, Kashmiri	Chemistry, Maths	Evening	Morning, Afternoon, Evening	Wednesday	Tuesday, Monday	Monday, Wednesday, Saturday
	Black Frieza	42	Maharashtra	Odia	Telugu, Assamese	Chemistry	Night	Afternoon, Night	Sunday	Thursday, Saturday	Monday, Wednesday, Friday
	Ayush Singh	68	Meghalaya	Tulu	Santali, Mewari	Maths	Afternoon	Morning, Afternoon	Wednesday	Thursday, Wednesday	Tuesday, Wednesday, Saturday
	Armin Arlert	49	Punjab	Laman/Lambadi	Sanskrit, Bengali	Maths	Morning	Afternoon, Night	Tuesday	Friday, Wednesday	Friday, Saturday, Sunday
	Eren Yeager	66	Telangana	Malvi	Maithili, Sanskrit	Physics, Maths	Night	Morning, Afternoon, Evening	Sunday	Wednesday, Friday	Monday, Tuesday, Wednesday
	Mikasa Ackerman	54	Arunachal Pradesh	Malvi	Sanskrit, Marwari	Physics	Night	Morning, Afternoon	Tuesday	Thursday, Sunday	Monday, Wednesday, Sunday
	Ram sir	38	Madhya Pradesh	Bengali	Maithili, Telugu	Physics	Evening	Afternoon, Night	Saturday	Friday, Wednesday	Tuesday, Saturday, Sunday
	Jay Aslaliya	59	Goa	Sanskrit	Urdu, Odia	Maths	Afternoon	Evening, Night	Saturday	Monday, Thursday	Tuesday, Thursday, Saturday
	Zenitsu Agatsuma	45	Uttarakhand	Santali	Marathi, Dogri	Maths	Morning	Afternoon, Evening	Tuesday	Wednesday, Tuesday	Tuesday, Wednesday, Thursday
	Grand Priest	62	Andaman and Nicobar Islands	Dogri	Mewari, Telugu	Physics	Morning	Morning, Afternoon, Evening, Night	Friday	Tuesday, Wednesday	Wednesday, Friday, Sunday
	Marco Bott	70	Madhya Pradesh	Hindi	Kannada, Assamese	Maths	Evening	Afternoon, Evening	Tuesday	Monday, Tuesday	Wednesday, Thursday, Saturday
	Giyu Tomioka	63	Kerala	Meitei (Manipuri)	Telugu, Marathi	Chemistry	Morning	Morning, Night	Sunday	Sunday, Thursday	Monday, Thursday, Sunday
	Aejaz Khan	63	Kerala	Mewari	Malayalam, Marwari	Chemistry	Evening	Afternoon, Evening	Saturday	Friday, Sunday	Monday, Tuesday, Thursday
	Khalid Ansari	46	Tamil Nadu	Marwari	Hindi, Odia	Physics, Maths	Night	Morning, Afternoon	Thursday	Saturday, Sunday	Monday, Wednesday, Saturday
	Hannes	29	Chhattisgarh	Odia	Assamese, Marathi	Physics, Chemistry	Night	Morning, Afternoon, Evening, Night	Tuesday	Saturday, Monday	Monday, Thursday, Friday
	Seema Koulkar	32	Tripura	Kannada	Marwari, Malvi	Physics, Chemistry, Maths	Morning	Morning, Evening	Wednesday	Thursday, Wednesday	Friday, Saturday, Sunday
	Ujwala Bharambhe	32	Chhattisgarh	Kannada	Kannada, Meitei (Manipuri)	Maths	Evening	Evening, Night	Friday	Thursday, Saturday	Tuesday, Wednesday, Friday
	Sanemi Shinazugawa	66	Telangana	Marwari	Marathi, Mewari	Chemistry	Afternoon	Evening, Night	Monday	Tuesday, Friday	Tuesday, Wednesday, Thursday
	Tanjiro Kamado	23	Madhya Pradesh	Dogri	Sanskrit, Marwari	Physics, Chemistry	Night	Afternoon, Evening	Wednesday	Tuesday, Saturday	Monday, Thursday, Saturday

Figure 4.2.1 Dataset Columns 1-11

	Preferred_Days-4	Preferred_Days-5	Preferred_Days-6	Preferred_Days-7
2	Monday, Tuesday, Wednesday, Sunday	Monday, Tuesday, Thursday, Friday, Saturday	Monday, Tuesday, Wednesday, Friday, Saturday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
3	Monday, Friday, Saturday, Sunday	Monday, Tuesday, Thursday, Friday, Saturday	Monday, Wednesday, Thursday, Friday, Saturday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
4	Monday, Tuesday, Wednesday, Friday	Monday, Tuesday, Wednesday, Friday, Saturday	Monday, Wednesday, Thursday, Friday, Saturday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
5	Tuesday, Thursday, Friday, Sunday	Monday, Tuesday, Wednesday, Friday, Saturday	Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
6	Tuesday, Friday, Saturday, Sunday	Monday, Wednesday, Thursday, Saturday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
7	Monday, Wednesday, Thursday, Saturday	Monday, Wednesday, Friday, Saturday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
8	Monday, Tuesday, Friday, Sunday	Monday, Tuesday, Wednesday, Thursday, Saturday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
9	Tuesday, Thursday, Saturday, Sunday	Monday, Wednesday, Thursday, Friday, Saturday	Monday, Tuesday, Wednesday, Thursday, Friday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
10	Tuesday, Thursday, Friday, Saturday	Monday, Wednesday, Thursday, Friday, Saturday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
11	Monday, Wednesday, Thursday, Sunday	Tuesday, Wednesday, Thursday, Friday, Saturday	Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
12	Monday, Thursday, Friday, Sunday	Monday, Tuesday, Friday, Saturday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
13	Monday, Wednesday, Thursday, Friday	Monday, Wednesday, Thursday, Saturday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
14	Wednesday, Thursday, Friday, Saturday	Wednesday, Thursday, Friday, Saturday, Sunday	Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
15	Tuesday, Thursday, Friday, Sunday	Tuesday, Wednesday, Thursday, Saturday, Sunday	Monday, Wednesday, Thursday, Friday, Saturday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
16	Tuesday, Friday, Saturday, Sunday	Monday, Tuesday, Thursday, Friday, Saturday	Monday, Wednesday, Thursday, Friday, Saturday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
17	Monday, Thursday, Saturday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday	Monday, Wednesday, Thursday, Friday, Saturday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
18	Monday, Thursday, Friday, Sunday	Tuesday, Thursday, Friday, Saturday, Sunday	Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
19	Monday, Wednesday, Thursday, Sunday	Monday, Tuesday, Friday, Saturday, Sunday	Monday, Tuesday, Wednesday, Friday, Saturday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
20	Monday, Thursday, Friday, Sunday	Monday, Tuesday, Wednesday, Thursday, Saturday	Monday, Tuesday, Wednesday, Friday, Saturday, Sunday	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday

Figure 4.2.2 Dataset Columns 12-15

Mode	Mode-2	ExperiencePhysics	ExperienceChemistry	ExperienceMaths
doubt solving	Video lectures,live lectures	0	6	38
doubt solving	Video lectures, one-on-one, live lectures	0	26	0
one-on-one	Video lectures, one-on-one, doubt solving	0	0	1
Video lectures	Video lectures,live lectures	0	0	42
one-on-one	Video lectures, one-on-one, doubt solving, live lectures	29	0	45
Video lectures	Video lectures, one-on-one	36	0	0
doubt solving	Video lectures, one-on-one, doubt solving	44	0	0
Video lectures	Video lectures, doubt solving	0	0	36
one-on-one	Video lectures, one-on-one, doubt solving	0	0	47
one-on-one	Video lectures, doubt solving, live lectures	23	0	0
Video lectures	Video lectures, one-on-one, doubt solving	0	0	40
Video lectures	Video lectures, doubt solving	0	20	0
Video lectures	Video lectures, doubt solving	0	25	0
doubt solving	doubt solving, live lectures	22	0	26
live lectures	Video lectures, one-on-one, live lectures	40	22	0
doubt solving	one-on-one, live lectures	10	12	3
live lectures	Video lectures, one-on-one, doubt solving, live lectures	0	0	14
Video lectures	Video lectures, one-on-one, doubt solving	0	9	0
one-on-one	Video lectures, doubt solving	46	40	0

Figure 4.2.3 Dataset Columns 16-20

4.2 Model Pipeline

1) First we take inputs from mentor which are :-

- Region
- Languages
- Preferred time
- Mode
- ExperiencePhysics
- ExperienceChemistry
- ExperienceMaths

2) Calculate Cosine similarity between user preferences and mentor details showing connection of interests

3) Reduce the dimensionality of features of the mentor from dataset by using PCA (Principal Component Analysis) Dimensionality Reduction method.

4) Pass the reduced dimension features to ANN (Approximate Nearest Neighbor) algorithm to get the nearest neighbors.

5) Return a list of most appropriate mentors to users according to similarity scores and other methods.

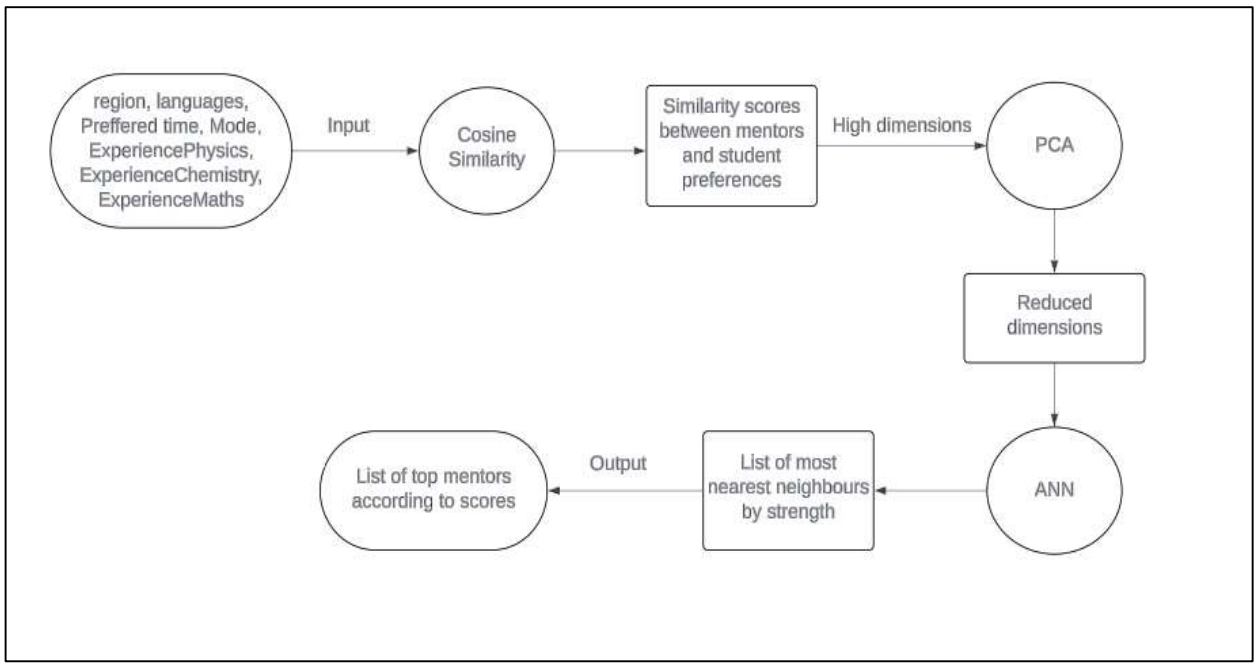

Figure 4.3 Model Pipeline diagram

4.3 Performance Evaluation

The successful implementation of mentor matching was made possible through the effective application of Natural Language Processing (NLP), a field of artificial intelligence that focuses on the interaction between computers and human language. By harnessing the power of NLP, the project was able to achieve remarkable results. Specifically, the utilization of the cosine similarity[4] metric in this context proved to be a game-changer. When applied to the extensive dataset, the results were nothing short of impressive. A maximum similarity score of 97.89 percent was attained in a critical test case, signifying a robust and substantial degree of similarity between the paired data points.

This exceptional outcome not only demonstrates the efficacy of cosine similarity as a measurement tool but also underscores the project's potential to revolutionize mentor matching, thereby benefiting countless individuals seeking guidance and mentorship. The use of NLP and advanced similarity metrics like cosine similarity[5] in this endeavor marks a significant advancement in the realm of mentorship programs and showcases the tremendous possibilities that lie ahead in creating more effective and meaningful mentor-mentee relationships.

4.4 Comparative Study

We conducted a detailed comparative study investigating the performance of four recommendation algorithms: Autoencoder, Cosine Similarity, Approximate Nearest Neighbor (ANN), and K-Nearest Neighbors (KNN). The primary objective was to identify the algorithm that delivers the best k-precision score.

The study evaluated the four algorithms using a dataset (not specified in the provided excerpt). The k-precision metric was employed to assess the effectiveness of each algorithm in recommending relevant items to users. K-precision measures the proportion of the top k recommendations that are truly relevant to the user's interests.

The provided bar chart (not included in the excerpt) indicates that the Approximate Nearest Neighbor (ANN) algorithm achieved the highest k-precision score of 0.90. This suggests that ANN outperformed the other three algorithms (Autoencoder, Cosine Similarity, and KNN) in generating highly accurate recommendations for the chosen value of k.

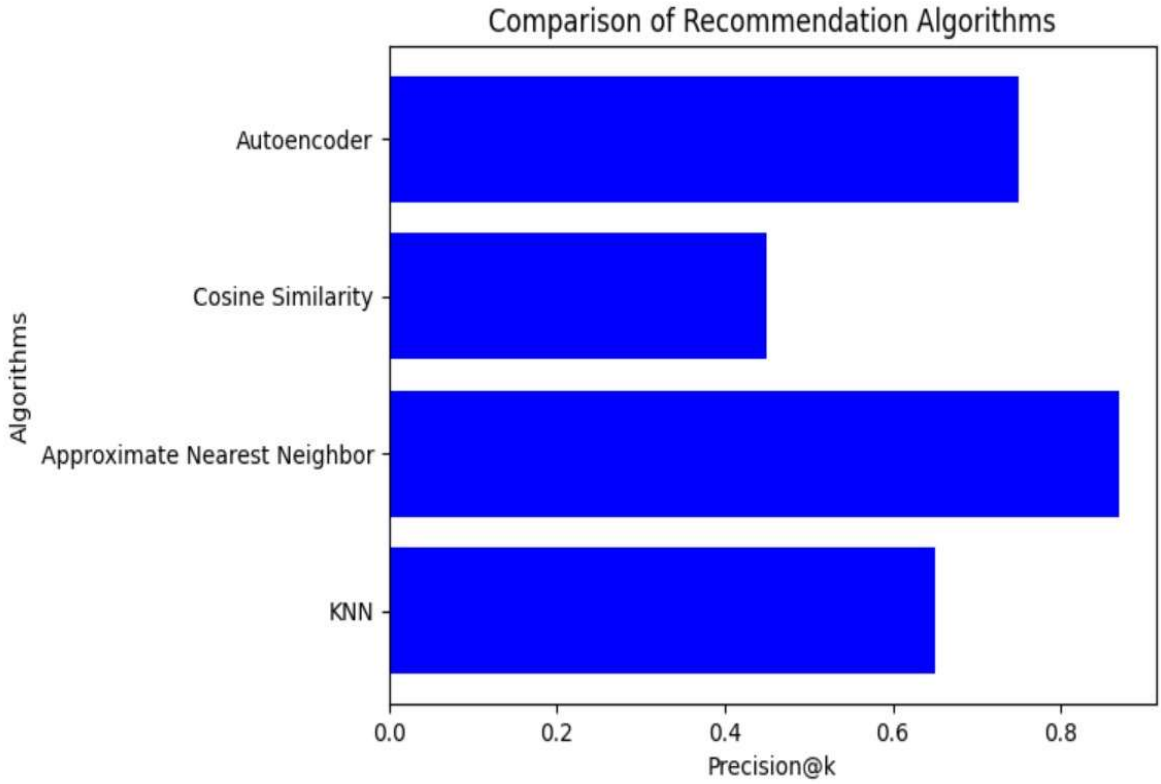

Figure 4.4.1 Comparative study between algorithms

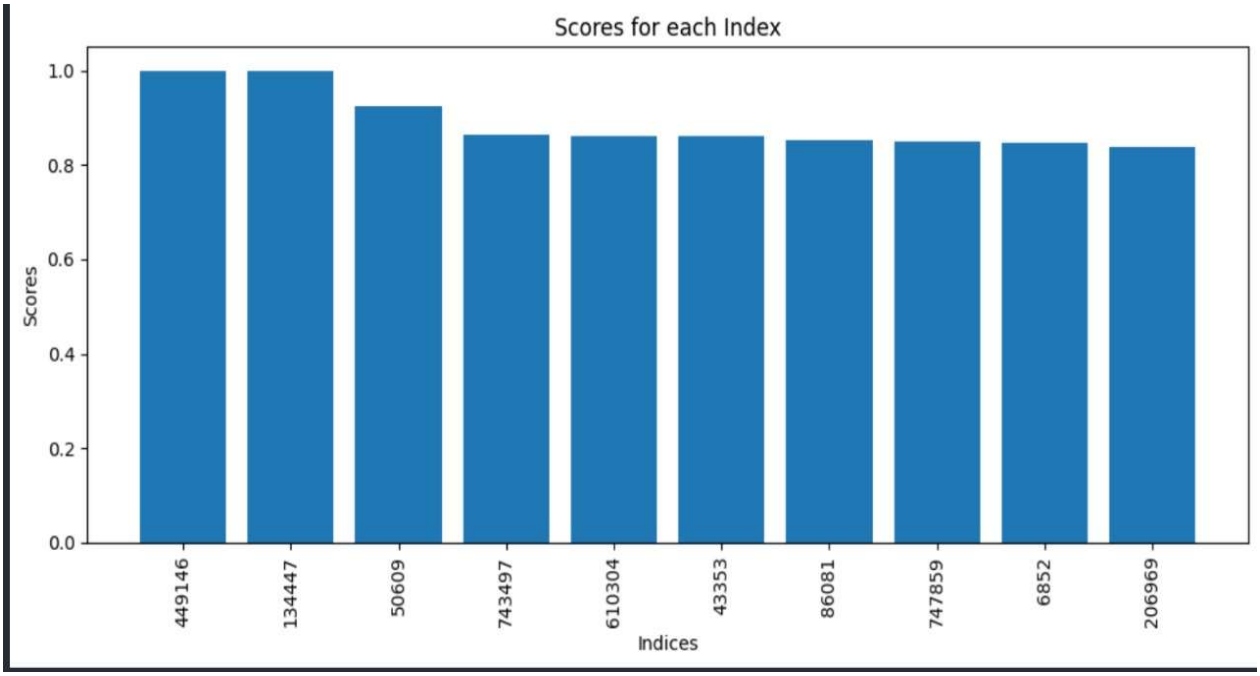

Figure 4.4.2 Mentors' recommendation scores graph

4.5 Software and Hardware Setup (Description of Libraries Used)

Python (3.10.4) : Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.
Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000. Python 3.0, released in 2008, was a major revision not completely backward-compatible with earlier versions. Python 2.7.18, released in 2020, was the last release of Python 2. Python consistently ranks as one of the most popular programming languages.

Data Analysis and Machine Learning Libraries:

NumPy (1.24.2) : NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

Pandas (1.4.2) : pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis/manipulation tool available in any language. It is already well on its way toward this goal. pandas is well suited for many different kinds of data. The two primary data structures of pandas, Series (1-dimensional) and DataFrame (2-dimensional), handle the vast majority of typical use cases in finance, statistics, social science, and many areas of engineering.

Matplotlib and/or Seaborn (0.11.2) : Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib. Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on top matplotlib library and is also closely integrated with the data structures from pandas. Seaborn

aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs so that we can switch between different visual representations for the same variables for a better understanding of the dataset.

Deep Learning Frameworks : scikit-learn (v 1.3.2) (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is a NumFOCUS fiscally sponsored project.

Successful implementation of mentor matching was possible using nlp. .The cosine similarity metric, when applied to the dataset, yielded a similarity score of (apne voh model mai jo accuracy ka number hoga) for a test case, indicating a strong degree of similarity. This analysis showcases the potential of cosine similarity for measuring the resemblance between data points

4.6 Input and Output

After inputting the system with the preferences of a student, the system initiates a multi-step process to provide the student with the most tailored mentorship experience. Firstly, the system employs a sophisticated algorithm to select the mentor who is the closest match to the student's unique needs and goals, considering factors like expertise, subject matter, and teaching style. Once the primary mentor is chosen, the system doesn't stop there. It proceeds to assess the compatibility and similarities between the selected mentor and the entire pool of available mentors. This is achieved by applying advanced cosine similarity measures, which quantify the degree of alignment between each mentor's strengths and the student's specific requirements. By extending this similarity analysis to multiple mentors, the system offers a list of the most suitable mentors beyond the primary choice, ensuring that the student benefits from a diverse range of mentorship options, each uniquely aligned with their academic and personal aspirations.

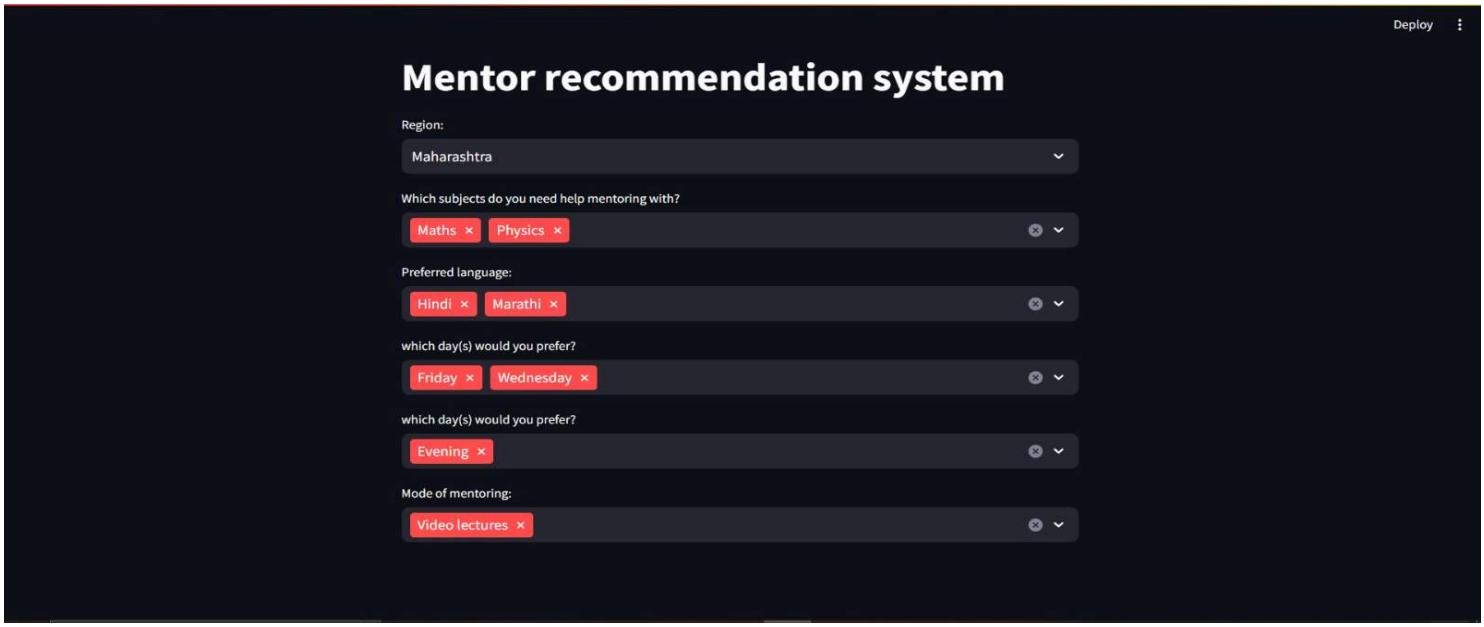

Figure 4.5.1 Input of Model

```
# Display recommended teachers
for teacher, score in recommended_teachers:
    print(f'Teacher ID: {teacher}, Similarity Score: {score}')

# You can add additional filtering and ranking logic in Step 6
```

```
[('Trunks Vegeta', 0.9999999999999998), ('Grisha Yeager', 0.9918225772359145), ('Son Goku', 0.9854467973922638), ('Reiner Braun', 0.9825425337356817), ('Akaza', 0.972743399372695), ('Manjiro Sano', 0
Teacher ID: Trunks Vegeta, Similarity Score: 0.9999999999999998
Teacher ID: Grisha Yeager, Similarity Score: 0.9918225772359145
Teacher ID: Son Goku, Similarity Score: 0.9854467973922638
Teacher ID: Reiner Braun, Similarity Score: 0.9825425337356817
Teacher ID: Akaza, Similarity Score: 0.972743399372695
Teacher ID: Manjiro Sano, Similarity Score: 0.9448125431609861
Teacher ID: Saiyan Cabba, Similarity Score: 0.9258681435150774
Teacher ID: Majin Buu, Similarity Score: 0.9184463012655513
Teacher ID: Kokushibo, Similarity Score: 0.8877100578737598
Teacher ID: Shinobu Kocho, Similarity Score: 0.8657208781749861
Teacher ID: Master Roshi, Similarity Score: 0.8549479666853799
Teacher ID: Attack Titan, Similarity Score: 0.8238786925256025
Teacher ID: Darakshan Khan, Similarity Score: 0.789590071031108
Teacher ID: Yoriichi Tsugikuni, Similarity Score: 0.770413250901238
Teacher ID: Eren Yeager, Similarity Score: 0.742779031808718
Teacher ID: Ujwala Bharambhe, Similarity Score: 0.7363425110350266
Teacher ID: Megumi Fushiguro, Similarity Score: 0.7196807105540761
Teacher ID: Tengen Uzui, Similarity Score: 0.6908676651433073
Teacher ID: Hinata Shoyo, Similarity Score: 0.6882542386475942
Teacher ID: Kale Caulifla, Similarity Score: 0.6316417776551524
Teacher ID: Reiner Braun, Similarity Score: 0.6143595062156231
Teacher ID: Mikasa Ackerman, Similarity Score: 0.5999818655468587
Teacher ID: Grand Zenos, Similarity Score: 0.5350118835643425
Teacher ID: Obanai Iguro, Similarity Score: 0.4956002556100074
Teacher ID: Gyomei Himejima, Similarity Score: 0.4764278416533816
Teacher ID: Ram sir, Similarity Score: 0.4752839846253627
Teacher ID: Itadori Kyuji, Similarity Score: 0.4409998179711574
Teacher ID: Ken Ryuguji, Similarity Score: 0.43698367543536254
Teacher ID: Tanjiro Kamado, Similarity Score: 0.43446934944166204
Teacher ID: Dot Pixis, Similarity Score: 0.42899439270042
Teacher ID: Muzan Kibutsuji, Similarity Score: 0.40083390486472337
Teacher ID: Grand Zenos, Similarity Score: 0.38348933045385564
Teacher ID: Gyomei Himejima, Similarity Score: 0.36888464964778866
```

Figure 4.5.2 Output of Model

Chapter 5

Implementation

5.1 Plan for Implementation

Week 1-2: Interface Design

- Design the user interface, incorporating transparency and real-time feedback features.
- Create wireframes and prototypes for the AI-driven decision interface.

Week 3-6: Development and Testing

- Begin developing the AI components, user interface, and real-time feedback mechanisms.
- Set up the database and backend infrastructure.
- Continue developing the system, focusing on personalization, bias detection, and mitigation.
- Implement educational resources and support modules. Ensure it aligns with the unique needs of the target user group.

Week 7-9: Testing and Quality Assurance

- Conduct comprehensive testing of the system, including functionality, usability, and security.
- Address any issues or bugs identified during testing.

Week 10-12: Deployment and Pilot Testing

- Deploy the system in a controlled environment for pilot testing.
- Gather feedback from real users, including IIT JEE aspirants in rural areas.

Week 13-16: Refinement and Iteration

- Based on feedback, make necessary refinements to the system.

Chapter 6

Conclusion

Hence, to conclude we did many things regarding the project. First of all we saw introduction about project. We saw in detail what it does and what it is, its aims and objectives, and Scope of the project. Then we saw literature review of the existing systems, their advantages and limitations. After that, in next section we saw our proposed system in detail which will try to overcome limitations of other systems along with its DFD(Data Flow Diagram). Then, we saw the implementation details that shows the things we have implemented. It gives details about dataset we created, the requirements and working of the model that we implemented. Later, in the end, we saw the timeline where we can see the detailed timeline and Gantt chart of the things we will implement in the next term.

References

[1] D. Xu, S. Yuan, L. Zhang and X. Wu, "FairGAN: Fairness-aware Generative Adversarial Networks," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 570-575, doi: 10.1109/BigData.2018.8622525.

[2] D. Xu, S. Yuan, L. Zhang and X. Wu, "FairGAN+: Achieving Fair Data Generation and Classification through Generative Adversarial Nets," 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, pp. 1401-1406, doi: 10.1109/BigData47090.2019.9006322.

[3] A. C. Braun, U. Weidner and S. Hinz, "Support vector machines, import vector machines and relevance vector machines for hyperspectral classification — A comparison," 2011 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Lisbon, Portugal, 2011, pp. 1-4, doi: 10.1109/WHISPERS.2011.6080861.

[4] M. Alobed, A. M. M. Altrad and Z. B. A. Bakar, "A Comparative Analysis of Euclidean, Jaccard and Cosine Similarity Measure and Arabic Wordnet for Automated Arabic Essay Scoring," 2021 Fifth International Conference on Information Retrieval and Knowledge Management (CAMP), Kuala Lumpur, Malaysia, 2021, pp. 70-74, doi: 10.1109/CAMP51653.2021.9498119.

[5] A. R. Lahitani, A. E. Permanasari and N. A. Setiawan, "Cosine similarity to determine similarity measure: Study case in online essay assessment," 2016 4th International Conference on Cyber and IT Service Management, Bandung, Indonesia, 2016, pp. 1-6, doi: 10.1109/CITSM.2016.7577578.

[6] https://www.khanacademy.org/

[7]https://www.coursera.org/courseraplus?utm_source=gg&utm_medium=sem&utm_campaign=B2C_INDIA__branded_FTCOF_courseraplus_arte_monthly&utm_content=B2C&campaignid=18216928761&adgroupid=141296026472&device=c&keyword=coursera&matchtype=b&network=g&devicemodel=&adpostion=&creativeid=619458216863&gclid=CjwKCAjw-eKpBhAbEiwAqFL0ms_-ApDh8xJJ6js9VeTyExXpOW4C2kcMksxBzmWHffcKll-SBovIHhoCifcQAvD_BwE

[8] https://byjus.com/

[9] https://www.dreambox.com

[10] https://www.smartsparrow.com/

[11] https://www.pw.live/

[12] https://github.com/jasonshere/FairGAN

[13] https://developers.google.com/machine-learning/recommendation/collaborative/basics

[14] https://developers.google.com/machine-learning/recommendation/collaborative/basics

[15] https://developers.google.com/machine-learning/recommendation/collaborative/basics

[16] https://blog.uniqin.ai/p/cosine-similarity-for-item-based#:~:text=Item%2Dbased%20filtering%20is%20calculated,similarity%20of%20the%20user%20affinity.

[17] https://towardsdatascience.com/using-cosine-similarity-to-build-a-movie-recommendation-system-ae7f20842599

[18] https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/#:~:text=SVM%20is%20a%20powerful%20supervised,work%20best%20in%20classification%20problems.

[19] http://scikit-learn.org/stable/modules/svm.html

[20] https://www.youtube.com/watch?v=lDwow4aOrtg&pp=ygUMc3ZtIHN0YW5mb3Jk

[21] https://www.youtube.com/watch?v=h9gpufJFF-0&pp=ygUaY29zaW5lIHNpbWlsYXJpdHkgc3RhbmZvcmQ%3D

Acknowledgement

We would like to express our gratitude and thanks to **Prof. Sakshi Surve** for her valuable guidance and help. We are indebted for her guidance and constant supervision as well as for providing necessary information regarding the project. We would like to express our greatest appreciation to our principal **Dr. G.T. Thampi** and head of the department **Dr. Tanuja Sarode** for their encouragement and tremendous support. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of the project.

2003002 Komalika Acharya

2003012 Jay Aslaliya

2003044 Vedant Deshmukh

2003185 Yash Vaskar