

# Personalized E-Learning Course Recommendations :

## A Chatbot Approach Using LangChain



Mavin Sao

Graduate School of Data Science,  
Chonnam National University,  
Gwangju, South Korea  
kr225043@jnu.ac.kr

\* Hoi-Jeong Lim

Graduate School of Data Science,  
Public Data Analytics Center,  
Chonnam National University,  
Gwangju, South Korea  
hjlim@jnu.ac.kr



# Content

---

1. Introduction
2. Challenge
3. Solutions
4. Methodology
5. Evaluation
6. Conclusion





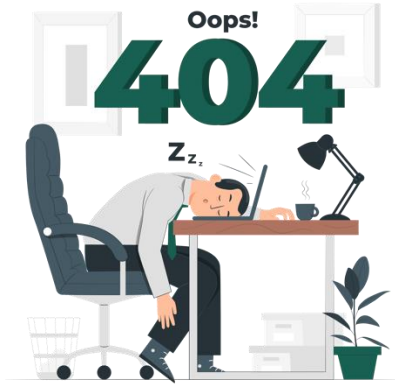
# Introduction



- Education has transformed significantly, especially during and after the Covid pandemic, **when students worldwide moved from traditional classroom learning to e-learning.**
- This shift has created a surge in online learning platforms like **edX**, **Coursera**, and others, making education more accessible and adaptable than ever before.



# Challenge



**Overwhelming**



**Time Consuming**



**Poor Selections**



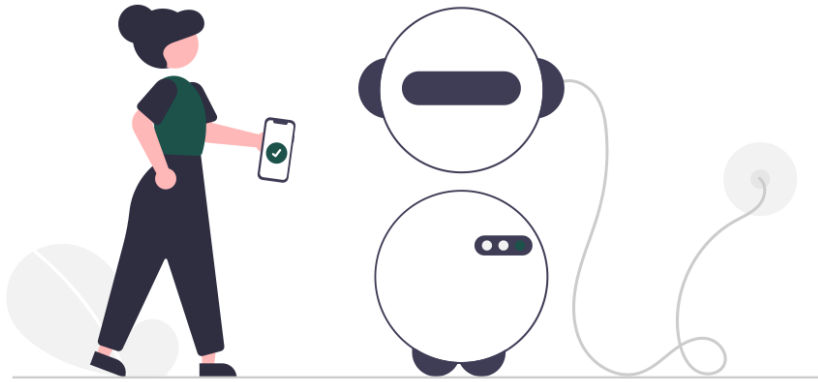
**Inaccurate & Hallucination**

**However, having so many options creates a major challenge:**

- Overwhelming number of online courses across platforms
- Time-consuming search process
- Difficulty matching courses to personal needs and goals
- Risk of making poor course selections
- Limited reliability of AI recommendations (like ChatGPT, Gemini, Claude) due to potential inaccuracies

# Solutions

**What if we leverage the capabilities of LLMs to recommend courses from the relevant courses based on their similarity to a given query?**



Based on your interest in learning Python for data manipulation and analysis, especially as a beginner in programming, I have tailored three introductory courses that will help you build a strong foundation. Each course focuses on essential skills, including the use of libraries like Pandas, which is crucial for data science applications.

**1. Fundamental Tools of Data Wrangling**

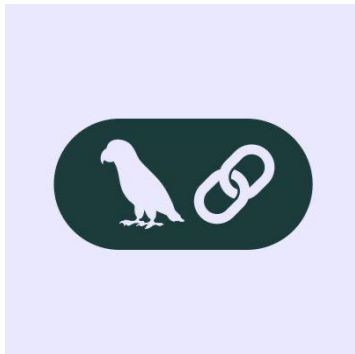
....

**A smart course recommendation system using AI and course data can help students quickly find the right online courses that match their needs, saving time and providing reliable guidance for their learning journey.**

---

# Methodology

To build such a course recommendation system, we need to understand several key technologies that work together.



- LangChain is an open-source framework that simplifies developing NLP applications using LLMs.
- It connects **LLMs to external data sources**, enabling powerful tools like question-answering chatbots through **Retrieval-Augmented Generation (RAG)**.

**Key elements of LangChain :**

**Vector Stores**

**Prompt Templates**

**Chains**

**LLMs**

**Agents**

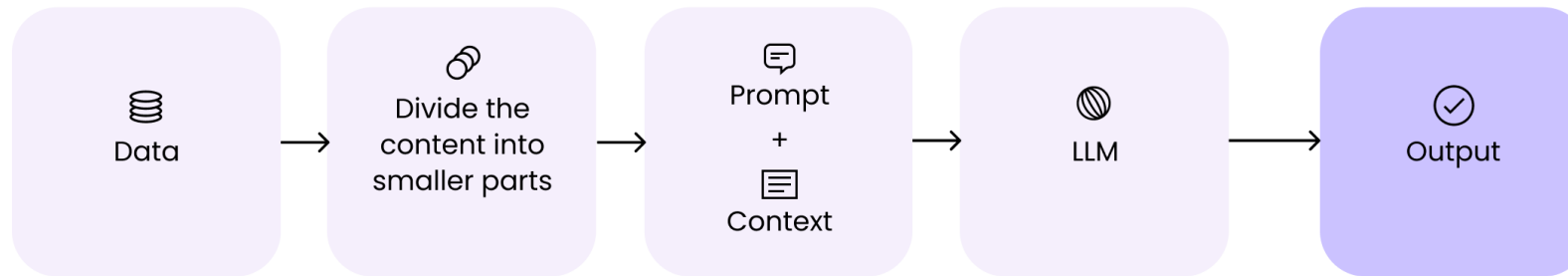
---



# Methodology

## What is Retrieval-Augmented Generation (RAG) ?

### The RAG process



RAG is a system that **looks at a large amount of data**, finds the **important pieces of content**, and **directs to a large language model as context**

---

Credit:

<https://writer.com/blog/retrieval-augmented-generation-rag/>

# Methodology

Below is our methodology for building our recommendation system. The overall pipeline process can be found in [Fig. 1](#).

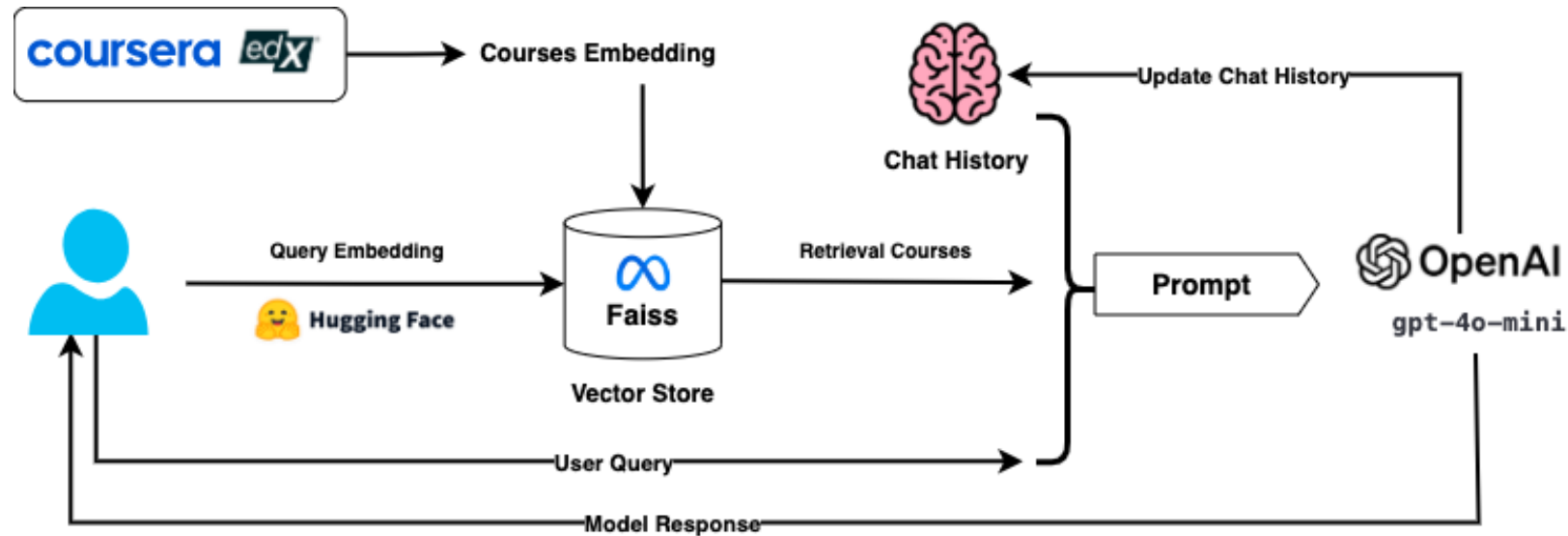


FIGURE 1. OVERALL PIPELINE OF THE COURSE RECOMMENDATION CHATBOT







# Methodology

## Data Collecting Process

**Input:** Topics (list of course topics to scrape)  
**Output:** CSV file containing course information

```
1: function ScrapeCourses(topics)
2:   webDriver ← InitializeWebDriver()
3:   allCourses ← empty list
4:
5:   for each topic in topics do
6:     totalPages ← GetTotalPages(webDriver, topic)
7:
8:     for pageNumber from 1 to totalPages do
9:       courseURLs ← GetCourseURLs(webDriver, topic, pageNumber)
10:
11:       for each url in courseURLs do
12:         courseInfo ← ExtractCourseInfo(webDriver, url)
13:         allCourses.append(courseInfo)
14:       end for
15:     end for
16:   end for
17:
18:   SaveToCSV(allCourses, "coursera_courses.csv")
19:   webDriver.close()
20: end function
```



- We used **Selenium WebDriver** for automated scraping of course information from **edX** and **Coursera**.
- Gathered course details such as: **Titles, Descriptions, Subjects, Ratings, Difficulty levels, Institutions, Objectives, Syllabi, URLs, and Skills outcomes**.
- We collected **16,223 courses** from **40 different subjects** which is an comprehensive amount and diverse subjects.

**FIGURE 2. WEB SCRAPING ALGORITHM FOR  
COURSE DATA EXTRACTION**



# Methodology

## edX Courses Webpage

Filter by

### Subject

- ☒ Data Science (990)
- ☐ Business (2,144)
- ☐ Computer Science (1,402)
- ☐ Information Technology (715)

[Show more](#)

### Language

- ☐ English (885)
- ☐ French (686)
- ☐ Arabic (672)
- ☐ Spanish (670)

[Show more](#)

### Learning Product

- ☐ Guided Projects (72)  
Build job-relevant skills in under 2 hours with hands-on tutorials.

Explore the Free Course Catalog

Data Science × [Clear all](#)



IBM IBM

#### Python for Data Science, AI & Development

**Skills you'll gain:** Computer Programming, Data Analysis, Python Programming

★ 4.6 · 39K reviews

Beginner · Course · 1 - 3 Months



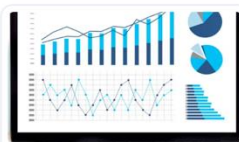
IBM IBM

#### Introduction to Artificial Intelligence (AI)

**Skills you'll gain:** Algorithms, Applied Machine Learning, Artificial Neural Networks, Computer Vision, Deep Learning, Human Learning, Machi...

★ 4.7 · 15K reviews

Beginner · Course · 1 - 4 Weeks



IBM IBM

#### Introduction to Data Analytics

**Skills you'll gain:** Data Analysis, Databases, Python Programming, SQL, Data Management, Data Mining, Data Science, Data Visualization,...

★ 4.8 · 17K reviews

Beginner · Course · 1 - 3 Months

All filters ×

Subject ^

Search 🔍

- ☐ Architecture
- ☐ Art & Culture
- ☐ Biology & Life Sciences
- ☐ Business & Management
- ☐ Chemistry
- ☐ Communication
- ☐ Computer Science
- ☐ Data Analysis & Statistics
- ☐ Design
- ☐ Economics & Finance
- ☐ Education & Teacher Training
- ☐ Electronics
- ☐ Energy & Earth Sciences
- ☐ Engineering
- ☐ Environmental Studies

Clear all

Apply



< 1 2 3 ... 20 >

Home > Learn > Python > HarvardX: CS50's Introduction to Programming with Python



## HarvardX: CS50's Introduction to Programming with Python

An introduction to programming using Python, a popular language for general-purpose programming, data science, web programming, and more.



10 weeks  
3-9 hours per week



Self-paced  
Progress at your own speed



Free  
Optional upgrade available

About

What you'll learn

Instructors

### About this course

An introduction to programming using a language called Python. Learn how to read and write code as well as how to test and "debug" it. Designed for students with or without prior programming experience who'd like to learn Python specifically. Learn about functions, arguments, and return values (oh my!); variables and types; conditionals and Boolean expressions; and loops. Learn how to handle exceptions, find and fix bugs, and write unit tests; use third-party libraries; validate and extract data with regular expressions; model real-world entities with classes, objects, methods, and properties; and read and write files. Hands-on opportunities for lots of practice. Exercises inspired by real-world programming problems. No software required except for a web browser, or you can write code on your own PC or Mac.

### At a glance

- **Institution:** [HarvardX](#)
- **Subject:** [Computer Science](#)
- **Level:** Introductory
- **Prerequisites:** None
- **Associated programs:** Professional Certificate in [Computer Science for Python Programming](#) Professional Certificate in [Learning Python for Data Science](#)

- **Language:** English
- **Video Transcript:** English
- **Associated skills:** Python (Programming Language), Computer Science, Web Browsers, Regular Expressions, Unit Testing, JavaScript (Programming Language), Boolean Expression, Web Development, C (Programming Language), Debugging, Data Science, SQL (Programming Language), Personal Computers

### What you'll learn

- Functions, Variables
- Conditionals
- Loops
- Exceptions
- Libraries
- Unit Tests
- File I/O
- Regular Expressions
- Object-Oriented Programming
- Et Cetera



# Methodology

## Coursera Courses Webpage

coursera



Filter by

Subject

- ☐ Business (2,144)
- ☐ Computer Science (1,402)
- ☐ Data Science (990)
- ☐ Information Technology (715)

[Show more](#)

Language

- ☐ English (6,718)
- ☐ French (5,402)
- ☐ Spanish (5,367)
- ☐ Portuguese (Brazil) (5,075)

[Show more](#)

Learning Product

- ☐ Guided Projects (547)  
Build job-relevant skills in under 2 hours with hands-on tutorials.
- ☐ Courses (6,419)  
Learn from top instructors with graded assignments, videos, and discussion forums.
- ☐ Projects (43)  
Learn a new tool or skill in an interactive, hands-on environment.
- ☐ Specializations (812)

Explore the Free Course Catalog



English for Common Interactions in the Workplace: Basic Level

★ 4.7 · 2.9K reviews  
Mixed · Course · 1 - 3 Months



Business Analysis & Process Management

Skills you'll gain: Business Analysis, Business Process Management

★ 4.4 · 5.5K reviews  
Beginner · Guided Project · Less Than 2 Hours



Build a free website with WordPress

Skills you'll gain: Creativity, Web Design, Web Development

★ 4.4 · 1.3K reviews  
Intermediate · Guided Project · Less Than 2 Hours



Introduction to Microsoft Excel



Financial Markets



First Step Korean

### Subject

Search for Subject

- ☒ Data Science (990)
- ☐ Information Technology (715)
- ☐ Social Sciences (567)
- ☐ Language Learning (199)
- ☐ Business (2,144)
- ☐ Health (705)
- ☐ Arts and Humanities (375)
- ☐ Math and Logic (78)
- ☐ Computer Science (1,402)
- ☐ Physical Science and Engineering (594)
- ☐ Personal Development (219)



### Python for Data Science, AI & Development

This course is part of multiple programs. [Learn more](#)

Instructor: [Joseph Santarcangelo](#)

Enroll for Free  
Starts Dec 9

Financial aid available

1,035,030 already enrolled

Included with [COURSERA PLUS](#) · [Learn more](#)

#### 5 modules

Gain insight into a topic and learn the fundamentals.

4.6 ★  
(38,959 reviews)

Beginner level  
Recommended experience

Flexible schedule  
Approx. 25 hours  
Learn at your own pace

95%  
Most learners liked this course

About Outcomes Modules Recommendations Testimonials Reviews

#### What you'll learn

- ✓ Learn Python - the most popular programming language and for Data Science and Software Development.
- ✓ Apply Python programming logic Variables, Data Structures, Branching, Loops, Functions, Objects & Classes.
- ✓ Demonstrate proficiency in using Python libraries such as Pandas & Numpy, and developing code using Jupyter Notebooks.
- ✓ Access and web scrape data using APIs and Python libraries like BeautifulSoup.

# Methodology

## Text Embedding Process

- **Sentence Transformers (SBERT)** is a Python library that provides easy access to **state-of-the-art text and image embedding models**, with over **5,000 pre-trained** models available on Hugging Face, including our chosen model **all-MiniLM-L6-v2**.
- The library enables various applications like semantic search and similarity scoring, while also allowing users to train or fine-tune custom models for specific use cases.

"The **weather** is lovely today."

"It's so **sunny** outside!"

"He drove to the stadium."



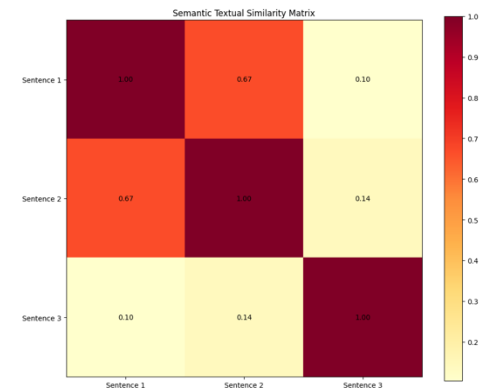
all-MiniLM-L6-v2



[1.91956814e-02, 1.20085381e-01, ....]

[-1.86903886e-02, 4.15186659e-02, ...]

[1.36501998e-01, 8.22731331e-02, ....]



Similarity Matrix



# Methodology

## Text Embedding Process

- The system uses the "**all-MiniLM-L6-v2**" Sentence Transformer to convert both course information and user queries into **384-dimensional vector embeddings**.
- This shared vector space enables direct similarity comparisons to generate recommendations.

```
# Initialize Sentence Transformer embeddings
model_name = "all-MiniLM-L6-v2"
embeddings = HuggingFaceEmbeddings(model_name=model_name)
```

```
# Create a vector store using FAISS
texts = df['combined_info'].tolist()

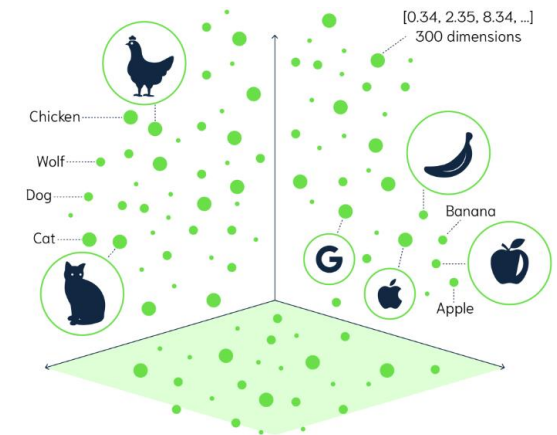
metadatas = df[['title', 'sub_info', 'rating', 'subject', 'level', 'institution', 'course_url']].to_dict('records')

# Create the vector store
vectorstore = FAISS.from_texts(texts, embeddings, metadatas=metadatas)
```

```
# Save the FAISS index
vectorstore.save_local("faiss_index")

print("FAISS index saved to faiss_index directory")
```

FAISS index saved to faiss\_index directory



Credit:  
<https://devocean.sk.com/blog/techBoardDe tail.do?ID=165867&boardType=techBlog>



# Methodology

## FAISS: Facebook AI Similarity Search

- **FAISS (Facebook AI Similarity Search)**, a library developed by Facebook, is designed for **measuring similarity and clustering of dense vectors**, serving as the **vector database in our RAG architecture to efficiently store and retrieve embedded course data**.
- By default, FAISS uses **L2 (Euclidean) distance** to measure vector similarity. It computes the L2 distances between the **query vector and all database vectors and returns the vectors with the smallest distances** (i.e., the most similar ones).

```
# Initialize Sentence Transformer embeddings
model_name = "all-MiniLM-L6-v2"
embeddings = HuggingFaceEmbeddings(model_name=model_name)

vectorstore = FAISS.load_local("faiss_index", embeddings, allow_dangerous_deserialization=True)
```

```
# Create a retriever from the loaded vector store
retriever = vectorstore.as_retriever(search_kwargs={"k": 5})
```

**Output 5 relevant documents**

```
# Method 1: Using retriever
docs = retriever.get_relevant_documents("""I'm an intermediate programmer with Python experience, looking to dive deep into machine learning
for financial applications.""")

print(docs)
```



# Methodology

## Prompt Template

- The recommendation system utilizes a structured prompt template that integrates retrieved course data, recommendation format, and chat history.
- This template guides the language model to analyze queries and generate personalized course recommendations while maintaining conversational flow.

```
# Define a prompt template for course recommendations
prompt_template = """
As an AI course recommendation expert, provide personalized, high-quality suggestions based on the user's interests, goals, and background.

Chat History: {chat_history}
User Query: {question}
Relevant Courses: {context}

Response Guidelines:
1. Tone: Warm, professional, and approachable.
2. Analysis: Consider user's query, history, and educational needs.
3. Recommendations: For each course, include:
   - Title and institution
   - Brief overview
   - Skills to be gained
   - Key topics
   - Level, duration, language
   - Ratings (if available)
   - Course URL (if available)
4. Personalization: Explain how courses align with user's interests and needs.
Prioritize accuracy, relevance, and user-centricity to help users make informed educational decisions.

Recommendation:
"""
```

```
PROMPT = PromptTemplate(
    template=prompt_template,
    input_variables=["chat_history", "question", "context"]
)
```



# Methodology

## Large Language Model with Memory

- Our recommendation system is **powered by the GPT-4o-mini model** for processing user queries and generating natural course recommendations.
- **ConversationSummaryBufferMemory** maintains a condensed conversation history within a **1000-token limit to preserve context across interactions**.
- By combining current requests with summarized past interactions, the system delivers personalized course recommendations that align with the user's evolving interests and needs.



GPT-4o-mini

```
PROMPT = PromptTemplate(  
    template=prompt_template,  
    input_variables=["chat_history", "question", "context"]  
)
```

```
# Initialize the language model  
llm = ChatOpenAI(temperature=0.7, model_name="gpt-4o-mini")
```

```
# Set up conversation memory with summarization  
memory = ConversationSummaryBufferMemory(llm=llm, max_token_limit=1000, memory_key="chat_history", return_messages=True)
```

```
# Create the conversational retrieval chain  
qa_chain = ConversationalRetrievalChain.from_llm(  
    llm=llm,  
    retriever=retriever,  
    memory=memory,  
    combine_docs_chain_kwargs={"prompt": PROMPT}  
)
```





# Methodology

## Streamlit Chatbot Interface

<https://academic-ally-course-recommendation-chatbot.streamlit.app/>

Try an Example Query

Select an example query:

Programming Courses

Try this example

**Academic Ally: Course Recommendation Chatbot**

Tailored course suggestions to suit your educational journey!

I want to learn programming. What courses would you recommend for a beginner?

Based on your interest in learning programming as a beginner, I have curated a list of courses that will provide you with a solid foundation in this field. Each course is designed to cater to newcomers, ensuring that you cover essential programming concepts and skills.

**1. Code Yourself! An Introduction to Programming**

- Institution:** Not specified
- Overview:** This course introduces the basics of programming using Scratch, a visual programming language designed for beginners.

What courses are you interested in?





# Evaluation Results



I'm an intermediate programmer with Python experience, looking to dive deep into machine learning for financial applications.



It's great to hear that you're looking to deepen your knowledge in machine learning for financial applications! Given your intermediate programming background in Python, you have a solid foundation to build on as you explore this exciting field.

Here are two course options that align well with your interests:

## 1. ACCA: Machine Learning with Python for Finance Professionals

- **Institution:** ACCA (Association of Chartered Certified Accountants)
- **Overview:** This course is designed specifically for finance professionals, focusing on practical skills in Python to implement machine learning in financial contexts. It provides a comprehensive introduction to machine learning concepts and their applications in finance.
- **Skills to be Gained:**
  - Practical Python programming for data analysis and machine learning
  - Ability to automate Excel workflows using Python
  - Understanding of machine learning models and improving their performance
- **Key Topics:**
  - Introduction to Python programming
  - Data analysis with pandas and visualization libraries
  - Automating Excel tasks using the xlwings library
  - Implementing real-world machine learning projects
- **Level:** Intermediate
- **Duration:** Not specified, but typically flexible with self-paced components
- **Language:** English
- **Course URL:** [ACCA: Machine Learning with Python for Finance Professionals](#)

## 2. UTAustinX: Fintech: AI & Machine Learning in the Financial Industry

- **Institution:** UTAustinX
- **Overview:** This course provides an overview of how machine learning is utilized in the finance industry, covering a range of applications from crowdfunding to robo-advising. It's part of a broader series on financial technology (Fintech).
- **Skills to be Gained:**
  - Understanding the applications of AI and machine learning in finance
  - Ability to model credit application decisions using Python
  - Knowledge of quantitative investments and social investing
- **Key Topics:**
  - Overview of the crowdfunding market
  - Basics of linear and non-linear machine learning models
  - Applications of machine learning in credit modeling, robo-advising, and quantitative investing
- **Level:** Intermediate
- **Duration:** Typically structured over several weeks, specifics may vary
- **Language:** English
- **Course URL:** [Fintech: AI & Machine Learning in the Financial Industry](#)

Both courses are well-suited to your goal of applying machine learning in finance, emphasizing practical skills in Python and relevant financial applications. If you need any more specific information about these courses or others, just let me know! Happy learning!

What are you looking to learn?



This structured presentation allows users to quickly compare the courses and access more information with a single click on the provided links.

The chatbot's response was evaluated through :

- **Automated metrics**
- **Human assessment**



# Evaluation Results

## Automated Metrics

To evaluate our chatbot system's performance, we analyzed **71 diverse queries** across various subject areas (**examples shown in Table I**).

We measured two types of semantic alignment using cosine similarity:

- **Query Relevance Score:** Between queries and responses - to check if answers matched the questions
- **Consistency Score:** Between retrieved context and responses - to verify if answers aligned with the source material

The similarity score for each query-response pair was computed using the following (1).

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} \tag{1}$$

TABLE I. EXAMPLES OF GENERATED QUERIES

Query	Category
I'm looking to enhance my leadership and organizational skills. Seeking courses in Business Management that cover strategic planning, team management, and operational efficiency.	Business Management
I want to learn programming from scratch. Looking for beginner courses in Python, Java, or C++.	Programming
I need to strengthen my understanding of data structures for better coding practices. Seeking courses that cover arrays, linked lists, trees, and graphs.	Data Structures
As an intermediate programmer, I'm looking to dive deep into machine learning for financial applications. Seeking advanced courses in Machine Learning and AI.	Machine Learning
I need to strengthen my statistical analysis skills for research. Seeking courses that cover probability, statistical inference, and regression analysis.	Statistics
I want to explore the laws of nature and physical phenomena. Seeking courses in Physics that cover mechanics, electromagnetism, and quantum physics.	Physics

TABLE II. AUTOMATED EVALUATION METRICS

Evaluation Metric	Avg. Score
Query Relevance Score	0.77
Consistency Score	0.77



# Evaluation Results

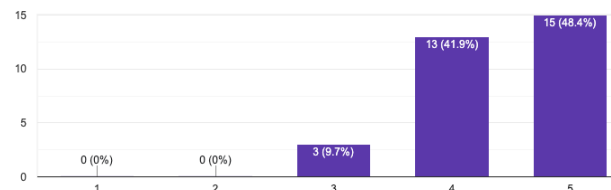
## Human Assessment Matrix

We conducted a human evaluation of the chatbot's responses through a survey that involved **31 students** from our graduate school.

How relevant was the chatbot's response to the given query?

[Copy chart](#)

31 responses

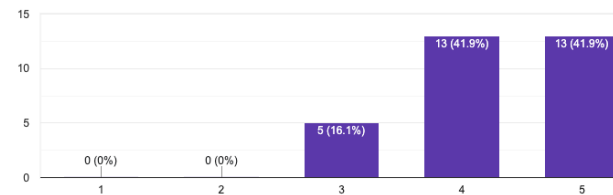


A. Relevance of Response to Query

How well did the chatbot's response format enhance readability?

[Copy chart](#)

31 responses

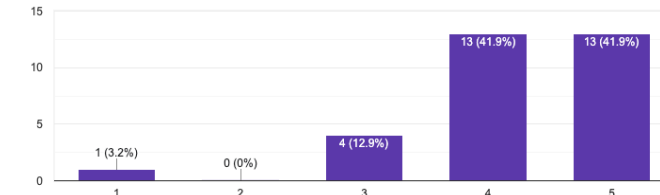


C. Response Format's Enhancement of Readability

How helpful was the explanation given for the course recommendations?

[Copy chart](#)

31 responses

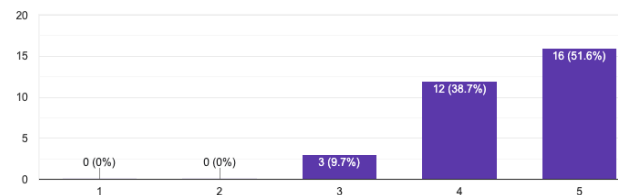


E. Helpfulness of Course Recommendation Explanation

How easy was it to understand the chatbot's response?

[Copy chart](#)

31 responses

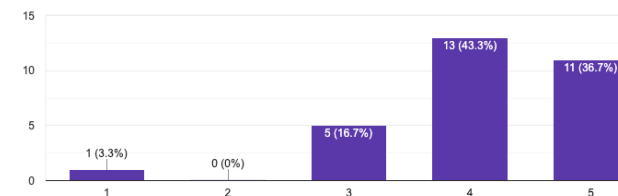


B. Ease of Understanding Response

How accurate were the course recommendations provided by the chatbot?

[Copy chart](#)

30 responses



D. Accuracy of Course Recommendations



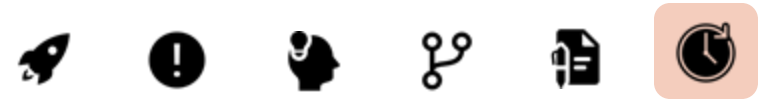
# Evaluation Results

## Human Assessment Matrix

- These scores indicate that the chatbot performed well across most criteria.
- The **ease of understanding (comprehensibility)** received the highest average score of **4.42**, followed by **relevance of responses at 4.39**.
- The **readability enhancement through response format** scored **4.26**, while the **helpfulness of explanations and accuracy of recommendations** scored **4.19** and **4.10** respectively.
- These results suggest that while the chatbot excels in delivering comprehensible and relevant responses, there might be **room for improvement in the accuracy of its course recommendations**.
- Nevertheless, all criteria received **scores above 4.0**, indicating **strong overall performance across all evaluated aspects**.

TABLE III. HUMAN EVALUATION RESULTS

Evaluation Metric	Avg. Score
Relevance of the chatbot’s response to the given query	4.39
Ease of understanding the chatbot’s response	4.42
Enhancement of readability through response format	4.26
Accuracy of the course recommendations provided	4.10
Helpfulness of the explanation given for course recommendations	4.19



# Conclusion

- This study developed a course recommendation chatbot using **GPT-4o-mini** and **LangChain**, with key features including **FAISS vector storage** and **RAG technology** to provide personalized course recommendations from **edX** and **Coursera data**.
- The evaluation was two-pronged:
  - **Human evaluation** showed strong performance (4.10-4.42/5.0) across relevance, comprehensibility, readability, accuracy, and explanation quality.
  - **Automated testing** focused on context relevance and consistency, achieving a 0.77 similarity score.
- The project demonstrated the effectiveness of combining large language model with structured data retrieval for educational recommendations.
- Future plans include exploring alternative vector stores (ChromeBD, Pinecone), enabling real-time updates, and developing a mobile application.

## ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MIST; RS-2023-00242528, 50%). This work was partly supported by the Innovative Human Resource Development for the Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MIST) (IITP-2024-RS-2022-00156287, 50%).

**Thank You**

# Q&A

