Course Recommender

Team:

- Oliver Pikani
- Joosep Tamm
- Mia-Liisa Kello

Github repo: https://github.com/Olivr/Course-Recommender

Business understanding

❖ Identifying your business goals

Background:

We've all been in a position where it's time to choose subjects for the next semester, however the choices are too broad and finding fun and interesting new subjects seems like an impossible task, especially when you want your topics to be connected.
Even if you know exactly what subjects you want to learn about and when you want to take the course, finding the right course can be very tedious and time consuming, especially for electives and optional courses, which can be taken from a much larger selection of courses (almost any course for optionals).

With this project we hope to make choosing courses to take at University of Tartu, especially courses from non-mandatory selections, much faster, simpler and more enjoyable.

Business goals:

Develop a LLM based GPT chat bot that has the knowledge of all the courses at University of Tartu and can answer questions and give suggestions about the courses.
Our main goal is to test the capabilities of the model and ensure it does not provide any false information.
Implement a suggestion feature that allows the chat bot to recommend specific courses based on user queries or preferences.
The general goal is to make course selection an easier process for all and increase the selection satisfaction for students.

Business success criteria:

The success and functionality of the eventual model will be measured subjectively by the model's users, initially that being us, the creators of the model. The criteria by which the model is assessed will be how relevant the model's response is to a prompt and how helpful the response is for actually planning the user's course selection.
The success criteria aligns with the overarching goal of providing relevant and helpful information about the courses at the University of Tartu, with a focus on real-world utility.

❖ Assessing your situation

Inventory of resources

The University of Tartu has provided us with access to OpenAI's GPT-3.5-Turbo model and embeddings API.
We also have our needed data on the courses at the University of Tartu thanks to SIS 2 API.
We are using Postman for API testing

Requirements, assumptions, and constraints

The model will need to be trained and be functional by December 11th.
This functionality requires the model to respond to prompts in a sufficiently accurate and relevant manner to truly help the user make decisions regarding course choice. This means that the model's responses should lead to the user finding a desirable course significantly faster than the current method, that being going through University of Tartu's website and manually going through all the courses offered in the desired field of study.

Risks and contingencies

One potential risk is going over the limits and costing the University of Tartu significant amounts of funds while training the model with GPT API and Embeddings API.
However, this is very unlikely and quite easy to avoid by calculating the token costs before making any requests to the APIs. If this problem somehow occurs despite making the calculations beforehand, then setting a token limit for each request would be the move to avoid any excessive token usage.

The second and largest potential issue is the model giving unhelpful or even actively distracting responses despite the availability of sufficient training data. This event would require changing how we synthesize data from the raw gathered data and mostly doing the whole project (data synthesis, model training and testing etc) again.

Terminology

- ★ **SIS** - Study Information System. The system that University of Tartu uses that contains all the information about courses

- ★ **LLM** - Large language model. A deep learning model that can understand, learn, summarize, translate, predict, and generate text and other content based on knowledge gained from massive datasets.

- ★ **GPT** - Generative Pre-trained Transformer. A type of machine learning model that's been pre-trained on a massive amount of data.

- ★ **GPT chat bot** - A chatbot that uses GPT to give textual answers to your prompts.

- ★ **OpenAI** - an artificial intelligence research company that produces services such as ChatGPT.

- ★ **Embeddings** - A process of converting high-dimensional data to low-dimensional data in the form of a vector in such a way that the two are semantically similar.

- ★ **API** - Application programming Interface. It's what software uses to access data, server software or other applications. In simple terms, it is a software intermediary that allows two applications to talk to each other.

- ★ **OpenAI API** - a collection of pre-trained AI models that enables us to integrate AI functionality into our applications without building and training our own models from scratch.

- ★ **API tokens** - Tokens can be thought of as pieces of words. Before the API processes the prompts, the input is broken down into tokens. 1 token ~= 4 chars in English.

Costs and benefits

The cost of 1K API tokens, using gpt-35-turbo, for the input is 0.0010$ and for the output 0.0020$.
The text-embedding-ada-002 usage costs 0.0001$ for 1K tokens.

Benefits of this project are the overall satisfaction of students and the chance to get more out of your university experience.

❖ Defining your data-mining goals

Data-mining goals

Data mining and synthesis should result in a minimal training set of data, based on which our model will be trained.

Data-mining success criteria

The data mining is considered successful when the following data can be retrieved:

- All data that is available in the SIS for each course
- Course's prerequisites
- All the curriculums where each course belongs to

Data understanding

❖ Gathering data

Data requirements

In order to train the LLM, data of all the courses offered at Tartu University is needed. This data must include:

- The names and descriptions of each course
- Which institute and field each course is a part of
- What language the courses are taught in
- When the courses are taking place (year and semester)
- The ECTS and given time expenditure of each course
- What other courses are prerequisites of each course and the alternative courses of those prerequisites
- The skills and information one should gain from the courses
- The study level (Bachelor's, Master's etc) of the courses
- The assessment scale (pass/fail or grading) of the courses
- The work assessments in the course (does it have an exam, homeworks, tests etc)

Verify data availability

All the required data is readily available on both Tartu University's webpage and its SIS2 (ÕIS2) learning system.

Define selection criteria

The data will be gathered from the SIS2 courses database, as this data is quite simple to gather through its API.

During the gathering process, all course data will be requested from the SIS2 API. All data fields we find are not required and all irrelevant datapoints are excluded from the final data.

❖ Describing data

The processed courses data is in JSON format with the following fields for each course:
"title" - name of course
"credits" - how many ECTS course is worth
"semester" - autumn or spring
"study_type" - partial or full-time
"faculty" - which faculty course is part of
"label" - contains when course is taking place and what part of a course it is
"type" - regular course, practice etc
"study_levels" - Bachelor's, Master's, applied etc
"hours" - how many hours each part (lectures, homeworks etc) are expected to take
"prerequisites" - what other course(s) are needed before that one can be taken
"study_languages" - languages the course is taught in
"description" - what the course is about
"objectives" - the purpose of the course
"learning_outcomes" - what a student taking the course should know by the end
"grading" - the course's grading system
"independent_work_assessments" - does the course have an exam, homeworks etc
"lecturer" - the faculty responsible for the course
"Curricula" - which curricula the course is a part of

All previously mentioned required data fields are present here, so the data is suitable.

❖ Exploring data

As all data is in text form and relatively unique to each course, there are no relevant distributions to mention and as the result of this project will be a trained LLM, there is no result hypothesis to make.

❖ Verifying data quality

The data can be accessed freely via their official API and therefore the dataset is clean and devoid of any artificial rows or entries.

Every datapoint is a genuine representation of a course offered by the University of Tartu and is therefore more than sufficient to proceed with training the model.

Before the actual API requests, the data which will be retrieved is evaluated using Postman

Planning your project

Task	**Description**	**Tools and methods**	**Hours**	**Assignee(s)**
Developing a project plan	Coming up with a initial plan for the project	Team meeting	5h	All team members
Research	Researching the necessary tools for the project	OpenAI's documentation on their APIs, any other useful site we find	5h	All team members
Gathering data	Gathering data from the SIS database	SIS2 API for courses, curricula etc	10h	Oliver
Synthesizing and formatting the data	Removing irrelevant data, refactoring row names and adjusting the data structure	Python scripts to iterate over the data	5h	All team members
Getting text-embeddings for each course	Using OpenAI's Embeddings API to retrieve embeddings for each course and storing them in a json file or database	Embeddings API. Stored in json file or vector database	5h	Mia-Liisa

Coming up with a good vector search algorithm	An algorithm to correlate query embeddings with course embeddings	K-nearest. Cosine similarity	5h	Joosep
Teaching GPT 3.5 Turbo	Providing GPT with test questions and answers and teaching it to answer the questions	GPT 3.5 Turbo API	3h	Oliver and Joosep
Testing model's competence	Testing the model with different queries and example data	Manual testing	3h	Mia and Oliver
Evaluating correctness of the model's answers	Evaluating GPT's answers and adjusting the model and input data accordingly	Manually evaluating the results	2h	Joosep and Mia
Further testing	Testing and improving the model	Manual testing	2h	All team members
(**bonus task, if we have extra time**) Gathering user feedback	Gathering user feedback and using the data to improve the relevancy of the answers to user queries	Implement upvote, downvote system for the program that reflects the accuracy of the answer.	2h	All team members
(**bonus task, if we have extra time**) GUI for the application	Develop a GUI for the program to be able to use it more easily	Browser interface	3h	Oliver