Group C11 report

Team members: Rannar Zirk, Anton Berik

Github link: https://github.com/RannarZ/DataScience_project

Business understanding

Background: As we are students of Tartu University and constantly wondering what suits our wishes and interests the most. But usually it is pretty difficult to find the correct courses as they are well hidden. For this reason we are looking for a way to make it easier for students to find courses based on their interests, skills and goals.

Business goals: Our goal is to simplify finding courses by creating an LLM model that will take any description of an user's interest in free form as an input and will return a course or courses which suit the best for the user's needs. Ideally this would fully replace the need to manually look for courses in the courses list in ÕIS II.

Business success criteria: This can theoretically be seen by the people who have taken the courses that the model gives them. Over 80% would be a success. Because sometimes people just look for courses without intention of taking them (In the scope of the IDS course we think that this test will not be possible but ideally this would be the success criteria).

Inventory of Resources: At our disposal we have the GPT 3.5 engine and ÕIS API for courses' information. There is also two of us who will run the project. We both also use our laptops and Jupyter Notebook for this project.

Requirements, assumptions, and constraints: For our project we have access to GPT 3.5 and GPT 4 model. We also have access to ÕIS API. Codes are kept in secret and inserted to the code from another file. It is also necessary for us to keep track of expenses. The project's deadline is for 11th of December.

Risks and contingencies: There are a few factors that might delay our work. One of them is obviously any power outages or internet connection problems. As we use Jupyter Notebook in our work then that requires network to work. Another thing that might delay our work is when the ÕIS API is undergoing an update and might be disabled for some period.

Terminology:
LLM – Large Language Model
GPT – Generative pre-trained transformer
API – Application programming interface

Costs and benefits: The GPT 3.5 model has some costs. For every 1000 tokens as input it costs $0.001 and as output $0.002. Text-embedding-ada-002 has a cost of $0.0001 for every 1000 tokens.

Data mining goals: Succesful use of the ÕIS courses API

Data mining success criteria: The only thing we can determine is the relevance of the info that the model generates. This will be done by a person (Rannar and Anton both) by creating some examples for which we want certain answers.

Data Understanding

Gathering Data:

Outline Data Requirements:
Our data requirements encompass three primary components:
* Student Profiles: Course details (year, degree), academic history.
* Course Information: Data on all available courses, content, and historical enrollment.
* Historical Enrollment Data: Records of past student course selections and outcomes.

Verify Data Availability: As we do not have access to Historical Enrollment Data, because of confidential policy, we are required to simulate those selections and outcomes. As mentioned before we have access to all of the courses and their requirements.

Selection Criteria: In our use we have the ÕIS II API from which we can gather information about any course that University of Tartu has to offer. For our project the most important parts are the courses names, descriptions, when they will be happening and what are the prerequisite courses.

Describing Data: While accessing our data we first saw that from our initial request we got the course's name, code, academic points and latest version. Which is almost everything we need. The only things missing are course descriptions and prerequisite courses. On initial search we didn't find them.

Exploring Data: On our first look through the data we didn't find any significant problems in the data. The data of courses consists of course name (in Estonian and English), code, uuid, state (which consists of code), last update, credits (points for the course), and the latest version's uuid.

Verifying Data Quality:
During our exploration, while we have not encountered explicit data quality issues there are some missing attribute that we might need. First one, as we mentioned before, was the course's description, which is not reachable via the ÕIS API (at least on initial research) and the prerequisite courses. There are no restrictions on existing data. We have access to everything that we need and that exists.

Project Plan for Course Recommendation System

Data Cleaning and Converting to Suitable Format:

* Tasks:
* Handle missing values, outliers, and inconsistencies.
* Standardize data format.
* Time Allocation:
* Rannar (0.5 hours).
* Anton (0.5 hours).

Algorithm Development:

* Tasks:
* Develop machine learning algorithms.
* Implement algorithms for personalized course recommendations.
* Time Allocation:
* Rannar (3 hours).
* Anton (3 hours).

GPT Model Implementation:

* Tasks:
* Integrate GPT model for natural language understanding.
* Train the GPT model.
* Time Allocation:
* Rannar (3h)
* Anton (3h)

User Interface Design:

* Tasks:
* Design an easy and user-friendly interface for students.
* Time Allocation:
* Rannar (1h)
* Anton(1h)

Testing and Validation:

* Tasks:
* Conduct thorough testing of algorithms and GPT model outputs.
* Validate the user interface for usability and responsiveness.
* Time Allocation:
* Rannar(0.5h)
* Anton(0.5h)

Methods and Tools:

Data Cleaning and Converting to Suitable Format:

* Methods: Utilize Python and Pandas for data cleaning and transformation.
* Tools: Jupyter Notebook

Algorithm Development:

* Methods: Implement machine learning algorithms using Scikit-learn.
* Tools: Scikit-learn for algorithm development, Jupyter Notebooks for experimentation.

GPT Model Implementation:

* Methods: Leverage pre-trained GPT models and fine-tune for specific educational data.
* Tools: GPT model given by teacher

User Interface Design:

* Methods: Follow prototype UX interface.
* Tools: Figma

Testing and Validation:

* Methods: Implement unit testing for algorithms
* Tools: Jupyter Notebook for algorithm validation.