EduGenie Final Report

Adaptive Applications (CS7IS5-202324)

School of Computer Science & Statistics

Team Poseidon

Members:
Mingyang Xia
Priya Radhamohan Nawal
Indrajeet Kumar
Xiaoyao Zhu
Alok Singh

April 15, 2024

Contents

1 Goal

Our goal with 'EduGenie' as an application is to architect a sophisticated, personalised university recommendation system that exploits the power of advanced machine learning algorithms and predictive analytics to streamline the university selection process. This platform meticulously analyses each university's admission standards and compares them with users' academic profiles to deliver finely tuned, customised recommendations. It is designed to identify the optimal universities that resonate with the users' personal aspirations and academic credentials, providing a calculated estimation of their acceptance odds.

2 Overview

"EduGenie" is an innovative platform designed to streamline the university selection process for a diverse array of prospective students, including recent graduates and international students seeking to study abroad. By leveraging cutting-edge Machine Learning technologies and employing robust data-driven analytics, this application conducts thorough analyses of extensive datasets containing detailed university profiles. This strategic approach facilitates the development of a sophisticated recommender system, intricately tailored to match the unique academic goals and personal preferences of each user. The system expertly identifies universities that align perfectly with each student's educational aspirations, significantly enhancing their chances of securing admission to their preferred institutions.

2.1 Key Features:

- **Personalized Recommendations:** Utilizing advanced algorithms, EduGenie analyzes users' academic backgrounds, interests, and prospective study programs to generate customized university suggestions. This ensures that each recommendation is uniquely suited to the user's specific educational goals.
- **Advanced Machine Learning Models:** The platform employs sophisticated ML techniques, including feature-weighted modelling and the K-Nearest Neighbors algorithm, to provide accurate and reliable university matches. These models effectively predict the best-fit universities based on a comprehensive analysis of user data and preferences.
- **User-Friendly Interface:** EduGenie's interface is designed for simplicity and ease of use, allowing users to effortlessly navigate through the recommendation process. This intuitive design helps users quickly access essential information about universities that meet their academic and personal criteria.
- **Data-Driven Strategic Selection:** By integrating user preferences with empirical data analysis, EduGenie enhances the likelihood of successful university admissions. The platform prioritizes selections from an elite group of universities, optimizing the match between student profiles and institutional offerings, thereby maximizing admission prospects.

Overall, EduGenie stands out as a pioneering solution in educational technology, offering a seamless and effective tool for making informed decisions about higher education opportunities.

3 Management Approach

The project was managed using a comprehensive suite of tools and platforms, each chosen for its specific utility in supporting various aspects of our workflow. Below is a detailed overview of the tools and platforms used during the project lifecycle:

- **Google Meet**: This tool was pivotal for internal communication. Weekly meetings were held every Wednesday at 11 am and Friday at 12 pm. These sessions were crucial for monitoring the progress of tasks, discussing strategic plans for upcoming sprints, and resolving any immediate issues that the team encountered.

- **Google Drive**: All project documents were meticulously stored and organized within a dedicated Google Drive folder. This setup provided the team with real-time access to necessary documents, facilitating seamless document tracking and sharing among team members. It proved instrumental in maintaining an organized and efficient document management system.

- **GitHub**: Our codebase was maintained on GitHub, which allowed for efficient code management and facilitated parallel collaboration across the team. The platform's robust version control capabilities enabled team members to review and merge changes, ensuring consistent feedback and substantially enhancing the overall quality of the code.

- **Jira**: We adopted an agile project management approach, utilizing Jira to manage our sprints and tasks. The project was structured into two-week sprints, enabling us to review and assess our progress at the end of each sprint. This regular evaluation helped identify areas for improvement and adjust role assignments based on workload planning. Jira was instrumental in documenting, planning, tracking, assigning, and managing tasks, which kept the project on track and well-organized.

The combination of these tools played a vital role in the successful management of our project. By leveraging each platform's strengths, we were able to maintain a high level of organization, communication, and productivity throughout the project life cycle. The agile methodology, supported by these digital tools, allowed for flexibility in project management and task execution, which was critical to the project's success.

4 User Modelling

In 'EduGenie', user modeling is an integral component that drives the precision and effectiveness of our recommendation engine. The process begins with an in-depth collection of user data through a detailed initial questionnaire. This questionnaire is meticulously designed to gather a wide range of information, including academic qualifications, personal preferences, language proficiency, standardized test scores, and desired fields of study. The rich data collected serves as the foundation for creating nuanced user profiles.

These comprehensive profiles are essential for the machine learning models to perform with high accuracy. They enable the system to understand the unique academic and personal characteristics of each user, forming a critical basis for tailored recommendations. By integrating sophisticated algorithms, these user profiles are analyzed with the existing database, containing diverse university criteria.

The algorithms employed are designed to discern patterns and matches that may not be immediately obvious, ensuring that each recommendation is not only based on surface-level compatibility but also on deep data-driven insights. This process ensures that the potential university matches suggested by 'EduGenie' are optimally aligned with each user's specific academic goals and personal aspirations, thereby significantly enhancing the likelihood of user satisfaction and success in their academic pursuits by not only provising them the list of suitable universities but also the chances in percentage of getting accepted to those universities.

5 Use Case

The University Recommendation System by 'EduGenie' meticulously matches students with suitable undergraduate and graduate programs, aligning personal preferences and academic profiles with institutional data. The platform offers customized recommendations to streamline the admissions process and maximize the prospects of applicants at their dream schools. The key application use cases are elaborated as follows:

- **Personalized University Recommendations for Aspiring Students:**This use case is crucial for students seeking comprehensive guidance in selecting universities for higher education. The system considers a wide array of factors, including academic performance, financial constraints, and personal aspirations. By inputting this information, students receive tailored suggestions that resonate with their unique preferences, significantly increasing their admission chances. It acts akin to a virtual adviser, empowering students in navigating their educational journey to find the most suitable institutions for their academic goals.

- **Efficient Graduate Program Selection:**Targeted at recent graduates or professionals eager to advance their education, this use case simplifies the selection of graduate programs. Users enter their academic credentials, such as GRE scores and undergraduate GPA, and the system provides recommendations for graduate programs that align with their qualifications and career ambitions. This functionality makes the decision-making process more efficient and streamlined, facilitating a smoother transition into further education.

- **Facilitating Cross-Border Education for International Students:**Designed to assist students planning to study abroad, this use case enhances the accessibility of international education. By specifying their preferred study destination and academic background, users receive personalized suggestions for colleges in their targeted region that offer programs meeting their specific needs. This feature simplifies the complexities associated with studying abroad and opens up new possibilities for cross-border educational experiences.

- **Supporting Academic Counsellors in Guidance Services:**Academic counsellors find immense value in this use case. It enables them to use the system during counselling sessions to input student's academic data and preferences. The algorithm processes this information and generates customized lists of suggested universities, which assists counsellors in providing individualized advice. This tool not only enhances the effectiveness of counselling sessions but also ensures students make well-informed decisions about their educational paths. It serves as an excellent resource for counsellors aiming to enhance their service quality and success in guiding students toward their academic objectives.

6 Architecture and Technology Stack

Our web application is elegantly constructed using a three-tier architecture model, ensuring a robust and scalable user experience from the ground up.

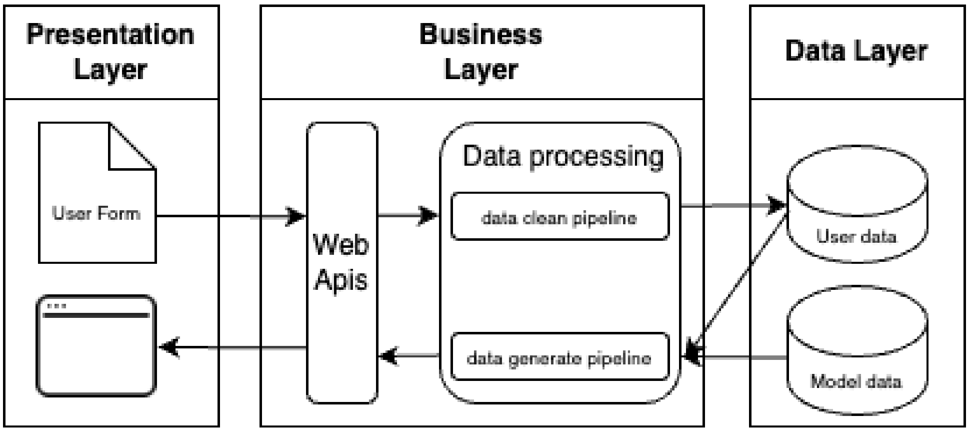

Figure 1: Functional Architecture of the System

6.1 Presentation Layer

The user interface, the gateway for user interaction, is built on the Next.js framework. This layer champions responsiveness and optimal content delivery through server-side rendering (SSR) and static site generation (SSG), ensuring that users receive a swift and dynamic experience. For aesthetics and functional design, we employed Tailwind CSS, which brings a utility-first philosophy to our HTML templates, enabling rapid, in-house customization of UI components while maintaining consistency and style.

6.2 Business Layer

The nucleus of our application, the Business Layer, operates on Python's versatile programming environment. This layer is the crux of our application's logic, from orchestrating web APIs to executing intricate algorithms. When tasks demand the prowess of machine learning or data-intensive analysis, TensorFlow is our tool of choice. It endows our application with the computational might to weave predictive models and insightful analytics into our core processes seamlessly.

6.3 Data Layer

Serving as the bedrock for our data operations, the Data Layer is powered by MySQL. This robust database management system provides steadfast reliability for storing, retrieving, and managing the intricate web of user and model data with remarkable efficiency, guaranteeing that our application's data-driven heart beats without interruption.

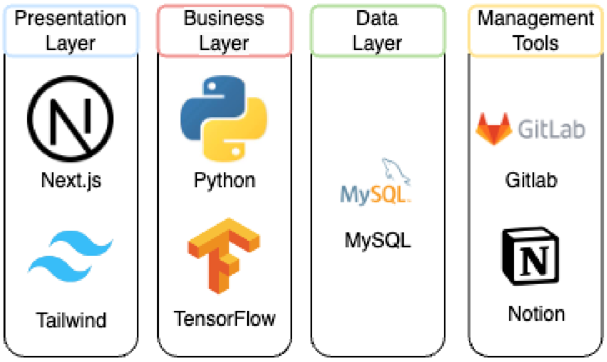

Figure 2: Technology Stack and Management Tools

6.4 Management Tools

Our development process is streamlined through an array of management tools. GitLab is the linchpin of our DevOps infrastructure, enhancing our software development lifecycle with its comprehensive integration and deployment pipelines. Complementing our workflow, Notion acts as our collaborative compendium, organizing documentation, tracking progress, and fostering team synergy.

This architecture and stack were not merely the backbone of our application but also the driving force behind our streamlined development process and our platform's capacity to adapt and scale. The harmonious integration of these technologies enabled us to build a system that is not only functional but also extensible, secure, and highly performant.

7 Implementation

The implementation of our university recommendation system is a testament to meticulous planning and strategic execution across various facets of development, from data preprocessing and machine learning to user interface design and adaptive features.

7.1 Support Vector Machine (SVM) Model Prediction

Support Vector Machine (SVM) is a powerful and versatile supervised machine learning algorithm, widely used for classification and regression tasks. SVM works by finding a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points [1].

7.1.1 Key Components

- **Hyperplane:** Decision boundary that separates different class labels.
- **Support Vectors:** Data points that are closer to the hyperplane and influence the position and orientation of the hyperplane.
- **Margin:** A gap between the two lines on the closest class points, maximized in SVM.

7.1.2 Formula

The decision function for SVM is defined as follows:

$$f(x) = w^T x + b \tag{1}$$

where w is the weight vector, x is the feature vector, and b is the bias. The optimization objective of SVM is:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \tag{2}$$

Subject to the constraints (for each labeled instance (x_i, y_i)):

$$y_i(w^T x_i + b) \geq 1 \tag{3}$$

7.1.3 Pseudo Code

The SVM algorithm can be summarized in the following steps:

1. Input: Training dataset $D = \{(x_i, y_i)\}$ where x_i is the feature vector and y_i is the class label ($y_i \in \{-1, 1\}$).
2. Initialize: Select an initial hyperplane.
3. Repeat:
   - Identify Misclassifications: Find points x_i that violate $y_i(w^T x_i + b) \geq 1$.
   - Update Rule: Adjust the hyperplane to correct misclassifications using:

$$w = w + \eta y_i x_i \tag{4}$$

$$b = b + \eta y_i \tag{5}$$

   - Optimization: Minimize $\frac{1}{2}\|w\|^2$ subject to the constraints.
4. Until Convergence or maximum iterations reached.
5. Output: The optimal hyperplane parameters w and b.

7.1.4 SVM Algorithm

SVM [2] uses a quadratic programming solution to solve the optimization problem. To make SVM capable of performing non-linear classification, we apply the kernel trick. The kernel function transforms the input data into a higher-dimensional space where a linear separator might be found. Common kernels include Polynomial, Radial Basis Function (RBF), and Sigmoid.

As we can see in figure 3, The process begins with Data normalization that ensure that the data is evened out for the model. The model is allowed to learn from the input data provided later allowing it to check its performance.

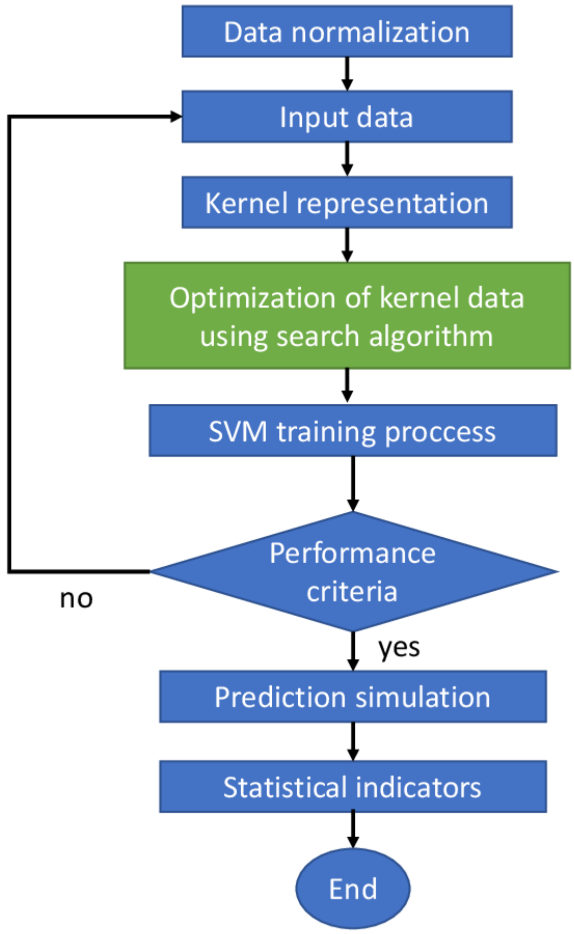

Figure 3: Flowchart of SVM model with Search algorithm

SVM [3] is a robust classification technique that works well on a wide range of classification problems, especially binary classification. It is effective in high-dimensional spaces and relatively memory-efficient. However, SVMs can be prone to overfitting when the number of features far exceeds the number of samples, which can be mitigated using appropriate regularization techniques [4].

7.2 Backend Implementation with Flask and Machine Learning

Our backend architecture, crafted with Flask, provided an efficient and structured foundation. Flask's flexibility was particularly advantageous for scripting and configuring the application. We leveraged Python for various scripting tasks, including data preprocessing, which involved tokenization and a series of other preprocessing steps. These processes were crucial for preparing the initial dataset, ensuring it was clean and structured, ready for the machine learning phase.

The preprocessed data then served as the input for training our machine learning models. We chose a Support Vector Machine (SVM) for its efficacy in classification tasks. The SVM model was trained on the processed dataset, fine-tuned to predict suitable universities for prospective students. To enhance our recommendation system's accuracy, we incorporated a Euclidean score calculation to measure similarities between user profiles and university attributes. This metric provided an additional layer of precision in our predictions.

For real-time user interactions, we developed an API that responds to user inputs by delivering a curated list of university recommendations along with the likelihood of acceptance. This functionality was embedded within the api.py file, which housed all the essential API endpoints, including the pivotal /submit endpoint.

7.3 User Interface Design

The UI of our application was crafted with user engagement in mind. At its core, the login and registration modules facilitated personalized access to the system's services. A comprehensive form submission interface allowed users to input personal and academic data, which the system used to generate and display a list of customized university recommendations on the schools page. To complement this data, we presented the acceptance rates and levels for each recommended university, along with a pie chart visualization to offer users an intuitive grasp of their personalized results.

7.4 Backend API Design

The backend API was meticulously designed to interface with the frontend seamlessly, ensuring that user requests for university recommendations were processed effectively. Our endpoints catered to both the retrieval of university data and the reception of user inputs, all handled with optimized security and performance.

7.5 Recommendation System

The recommendation system lies at the heart of our platform, combining user-centric design with advanced algorithms for a tailored list of universities, reflecting our dedication to a versatile and inclusive educational tool.

7.6 Adaptive Approach

Emphasizing a universal design, our application boasted an adaptive layout that responded to various devices and screen resolutions, ensuring a consistent experience across desktops, tablets, and smartphones.

In addition to this responsive design, we implemented a multi-language feature to cater to a global user base. Utilizing react-i18next, the application dynamically switched languages based on the user's browser settings, URL query parameters, or stored preferences [5]. This feature supported English, Hindi, and Chinese, showcasing our commitment to inclusivity and accessibility.

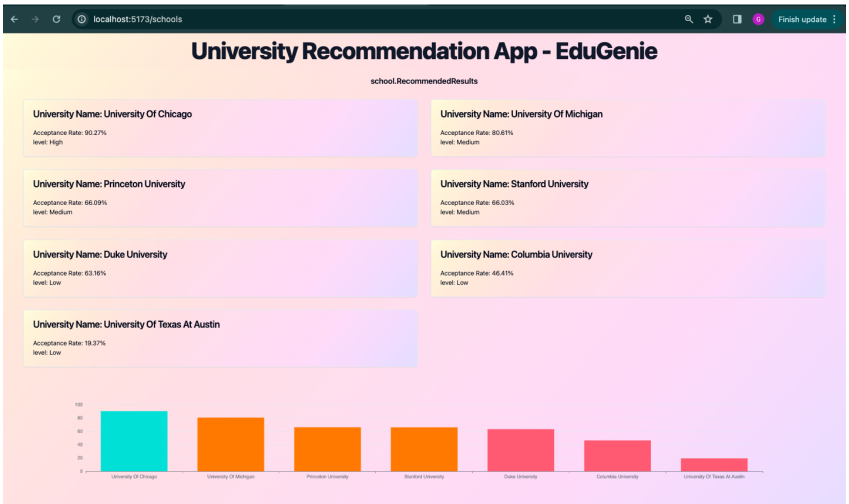

Figure 4: User Interface in English

Figure 5: User Interface in Hindi

Figure 6: User Interface in Mandarin

8 Conclusion and Future Work

In conclusion, our 'EduGenie' university recommendation system represents a significant stride in facilitating the educational journey of prospective students. By integrating advanced machine learning algorithms with a user-friendly interface and responsive design, we have created a tool that not only simplifies the university selection process but also personalizes it to meet the individual needs of users. The successful implementation of multilingual support further underscores our commitment to inclusivity and global reach.

As we look to the future, there are several avenues for further development. We plan to enhance the system's predictive accuracy by incorporating more nuanced user data and expanding our university database to include a broader spectrum of global institutions. Additionally, we aim to refine our language support to encompass more languages and dialects, ensuring that 'EduGenie' becomes even more accessible to students around the world. In terms of user experience, we will focus on integrating interactive elements, such as virtual campus tours and real-time chat with university representatives. On the technical side, exploring the integration of natural language processing (NLP) to interpret free-form user inputs could provide an even more tailored experience. By continuously iterating on feedback and staying abreast of technological advancements, we will strive to maintain 'EduGenie' at the forefront of educational technology and university matchmaking services.

References

[1] G. Baudat and F. Anouar. Generalized Discriminant Analysis Using a Kernel Approach. *Neural Computation*, 12(10):2385–2404, 2020.

[2] Research employing SVM for predicting student success in academia. [Online; accessed 20-October-2021].

[3] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 137–142, 1998.

[4] José Alvarez Alvarado, G.J. Moreno, Saul Obregón-Biosca, Guillermo Ronquillo, Eusebio Ventura-Ramos, and Jr Trejo-Perea. Hybrid Techniques to Predict Solar Radiation Using Support Vector Machine and Search Optimization Algorithms: A Review. *Applied Sciences*, 11(3):1044, 2021.

[5] lameer, A., Mahajan, S., & Halfond, W. G. J. (2016). *Detecting and localizing internationalization presentation failures in web applications*. In 2016 IEEE International Conference on Software Testing, Verification and Validation (ICST) (pp. 202–212). IEEE.