**Admission Compass**

Ariela Epstein- 328606686

Tal Uzan- 204344592

https://github.com/ariEppy/admissionCompassProject

**Table of Contents**

1. Introduction
2. Goals and Objectives
3. Functionality Requirements
4. Non-Functionality Requirements
5. Detailed Design
6. Literature Review
7. Competitive Analysis
8. Metrics
9. Algorithms
10. User Interface Design
11. Database Design
12. Use Cases
13. Conclusion

**1. Introduction**

Starting the journey of higher education can be intimidating and overwhelming. Students face challenges in making decisions about career paths, finding the right schools, and navigating through a myriad of admission requirements. Our project, Admission Compass, aims to support students by providing clear, accessible information about different admission requirements available at each school and degree, empowering future students with knowledge and easing their journey into higher education.

**2. Goals and Objectives**

**Goals:**

* Ensure access to information about available ways of admission.
* Empower students with the knowledge of different admission requirement possibilities.

**Objectives:**

* Develop a user-friendly website to help users explore degree programs and their admission criteria.
* Create a platform for accepted university students to share their admission criteria.
* Implement a search tool for users to refine searches by degree, institution, and personal scores.
* Utilize built-in tools, functions, and machine learning to present data analytics and predict future acceptance probabilities.

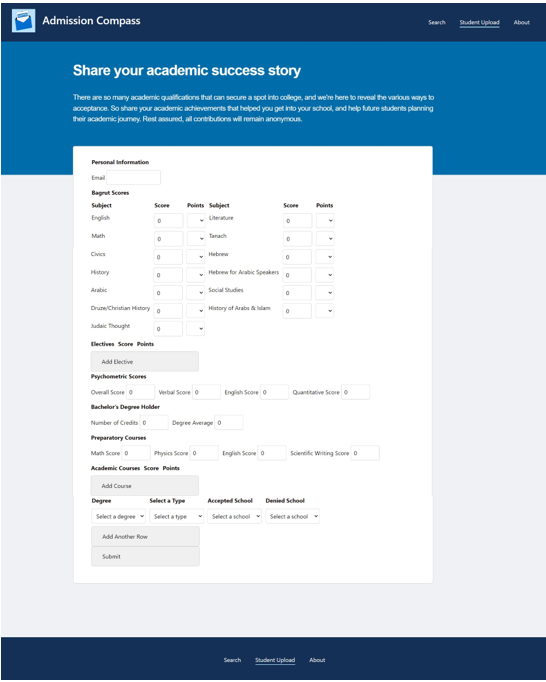
**3. Functionality Requirements**

**Admission Compass Features:**

1. **Accepted Students Admissions Form:** This form is designed to gather comprehensive academic achievements from accepted students, shedding light on the factors that contributed to their successful admission to higher education institutions. Users are invited to input a range of academic credentials, including Bagrut scores with specified electives, Psychometric scores, previous Degree averages (if applicable), Preparatory courses, and academic courses taken.

By detailing their accepted and denied school choices, along with respective degrees and degree types, students provide invaluable insights into the acceptance process. This information serves as the foundation for our predictive models, which analyze patterns and discern correlations between academic achievements and admission outcomes.

To ensure accuracy and compatibility with our existing database, our form employs intuitive dropdown menus for selecting degrees, degree types, and accepted or denied schools. Numeric inputs are required for all other score fields, ensuring data integrity and facilitating seamless integration with our backend systems.

Behind the scenes, our platform queries the database to retrieve the data for the dropdown menus, ensuring an up-to-date and comprehensive selection of schools and degrees.

1. **Search Tool – by School & Degree:** This powerful tool enables visitors to explore the diverse admission requirements offered by various schools and degrees. They can seek information on specific schools or degrees, and degree types here. Using intuitive dropdown menus, visitors can effortlessly navigate through available school, degree, and degree type options. This streamlined interface ensures a user-friendly experience while minimizing errors. We ensure accurate querying of the database and eliminate the risk of misspelled or incorrectly written entries. Behind the scenes, our tool dynamically retrieves data for dropdown menus through seamless queries to and from the database. This ensures that visitors have access to the latest information, enabling them to make well-informed decisions about their academic journey.

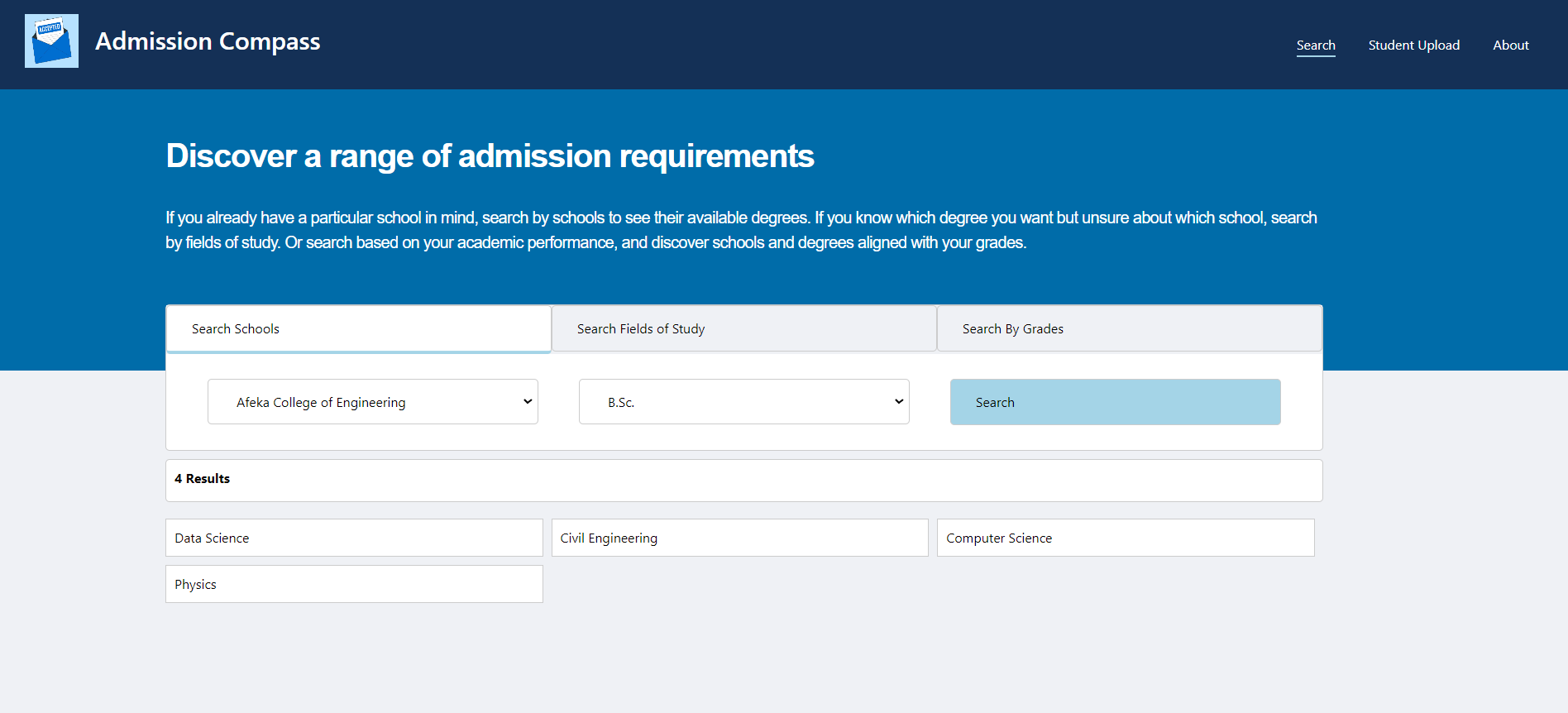
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **List of Search Results:** Upon submitting the search query, our intuitive search tool swiftly communicates with the database to retrieve relevant results based on the user's specified criteria, including the interested school, degree, and degree type. These results are promptly displayed below the search interface, providing users with a comprehensive overview of available options that match their preferences.

During this phase of website development, we have carefully curated a selection of universities and degrees to showcase on the site. While this initial sampling offers a glimpse into the wealth of possibilities, we are continuously expanding our database to include an even broader range of institutions and programs.

A screenshot of a computer

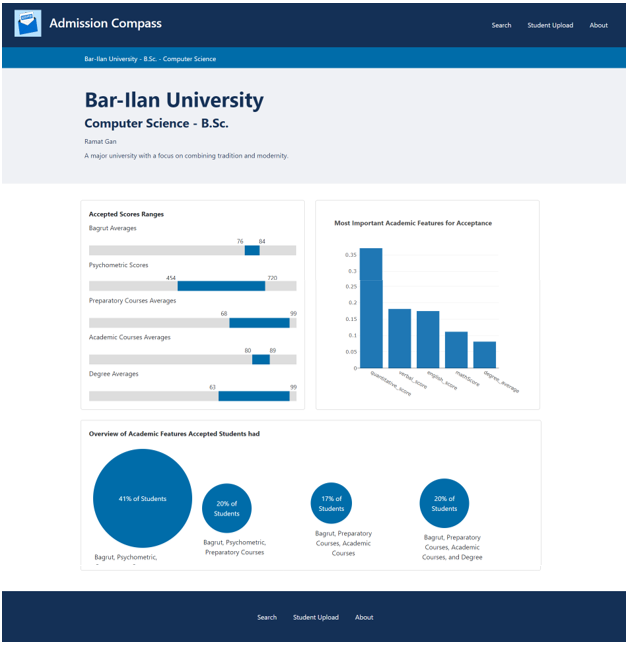
Description automatically generated

1. **The Detailed Results Page:** This page serves as a comprehensive showcase of the data stored in our database, providing visitors with valuable insights into the academic criteria that contribute to acceptance at various schools, degrees, and degree types.

Displayed on this page are the ranges of scores attained by accepted students at specific schools, degrees, and degree types, which rendered them eligible for acceptance. These scores span from 0 to 100, with psychometric grades ranging from 200 to 800. Additionally, we present combinations of academic features that led to acceptance into specific schools and programs, along with the corresponding percentages of students who possessed each combination. These percentages are calculated by dividing the number of students with a particular combination of academic achievements by the total number of students at the respective school, degree, and degree type. This enables visitors to discern the most prevalent academic feature combinations and understand the practical criteria for acceptance at each school.

The results showcased on this page are generated using MySQL's built-in mathematical functions, such as AVG, MIN, MAX, and FLOOR. Moreover, we leverage machine learning techniques to derive SHAP (SHapley Additive exPlanations) values from our dataset, shedding light on the academic features that significantly influence college acceptance. These SHAP values, representing features associated with acceptance (and not rejection), are presented in descending order of importance.

For instance, consider Bar Ilan University's B.Sc. program in Computer Science (in the picture below). Our analysis reveals that a significant proportion of accepted students (41%) were admitted based solely on Bagrut and Psychometric scores. Another substantial percentage (37%) gained acceptance through a combination of Bagrut scores, completion of preparatory courses, enrollment in academic courses, and holding or not holding a previous degree. The ranges of scores attained by accepted students at Bar Ilan University are as follows: Psychometric scores range from 454 to 720, preparatory course scores range from 66 to 99, Bagrut scores range from 76 to 84, academic course scores range from 80 to 89, and previous degree averages range from 63 to 99.

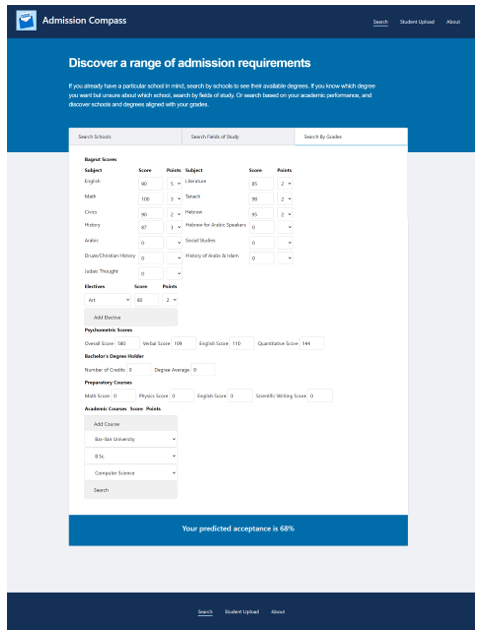
Moreover, our SHAP values indicate that Bar Ilan University places significant importance on quantitative scores in psychometric exams, followed by English and verbal scores. Additionally, a strong emphasis is placed on math preparatory course scores and students holding prior degrees. This information underscores Bar Ilan's preference for applicants with strong academic credentials, whether demonstrated through psychometric scores or completion of preparatory and academic courses.

1. **Search Tool - By Grades**: This page empowers visitors to gauge their likelihood of acceptance to a specified school, degree, and degree type by inputting their academic scores across various courses and tests. Users have the flexibility to enter Bagrut scores with specified electives, Psychometric scores, previous degree averages, completion of preparatory courses, and academic course grades.

To facilitate accurate predictions, dropdown menus containing school names, degree names, and degree types are dynamically populated from our database. This ensures alignment with our dataset and enhances the prediction process. Furthermore, all other input fields are numeric, ensuring compatibility with database values and minimizing errors.

Upon entering their academic data, users receive a predicted acceptance probability ranging from 0 to 100%. This prediction is generated using machine learning models trained on our dataset, which comprises data from accepted students. By leveraging multiple models, we enhance the accuracy of the prediction, providing users with a more reliable estimate of their acceptance likelihood.

For example, consider a student calculating their probability of acceptance to Bar Ilan University's Computer Science B.Sc. program. This student possesses a high Bagrut average, an average Psychometric score, excelling particularly in the math portion while scoring average to low marks in other sections. With no additional academic achievements to input, their predicted acceptance probability is 68%. This prediction may be attributed to the significance of the math portion in the psychometric exam, as identified earlier as a crucial factor for acceptance at this school and degree.



1. **Database Tables:** Our database serves as the cornerstone of our operations, storing data collected from accepted students via the website and providing data to the website based on user selections. It plays a pivotal role in generating analyses and predictions.

The database architecture consists of several interconnected tables, each serving a specific purpose. The "student submission" table acts as the central repository for general data about accepted students, linking to other tables that hold detailed information. These include the "bagrut," "psychometric," "academic courses holder," "degree holder," and "preparatory courses" tables.

When a student submits their data, it populates these tables sequentially, ensuring structured and organized storage of information. Additionally, our database includes tables such as "academic courses," "bagrut electives," "degrees," and "institutions," which store data related to courses, degrees, and institutions. This data is utilized to populate dropdown menus on the website, facilitating user interactions.

To maintain data integrity and security, we manually export relevant data from these tables for use in our machine learning models. This manual process helps ensure that the data collected from accepted students remains clean and free from potential online threats, safeguarding the integrity of our models.

1. **Machine Learning Models:** In Admission Compass, we embarked on training multiple machine learning models to predict the probability of a student's acceptance to a particular school and program. The process began with the extraction of the dataset from our database, ensuring its accuracy and integrity. As real-life data from students was not yet available, we initially generated data with random parameters.

Following this, we meticulously cleaned the dataset, ensuring that all data used for training was in a numeric format. Subsequently, we conducted thorough data analysis to identify correlations and patterns. To optimize model performance, we excluded features with minimal variation between entries, as they contributed little to the predictive capabilities of the models.

Our training phase encompassed six different models, each rigorously tested to evaluate their performance. After thorough evaluation, we identified the top-performing models: Random Forest, Decision Trees, and Gaussian Naive Bayes, which collectively yielded an accuracy rate of 80%.

Moreover, we leveraged machine learning models to provide insights to our visitors regarding the most influential factors for acceptance into various colleges and degrees. Utilizing SHapley Additive exPlanations (SHAP), we explained the outputs of our machine learning models by computing SHAP values. These ML tools serve as indispensable assets in enhancing the utility of our site for prospective students.

**4. Non-Functionality Requirements**

**User Experience:**

* Easy-to-use interface with straightforward navigation and clear instructions.
* Minimalist yet visually pleasing design to avoid overwhelming users.
* Information displayed clear and concise.

**Performance:**

* Quick responses to user queries, ensuring prompt delivery of search results and interactions.

**Scalability:**

* Our system has robust scalability and high user capacity, thanks to its adept handling of incoming user data through a specialized interface. Moreover, the site is scalable, since it enhances its performance as we get more users. Notably, our models continually improve in accuracy as they aggregate more data over time.

**Maintainability:**

* Clean and organized code.
* Continuous testing and monitoring for high performance and reliability.
* Efficient database management, regular updates, and new feature additions.

**5. Detailed Design**

**Physical Architecture:**

* **User Interface:** Users engage with the frontend interface via web browsers across various devices, including computers, phones, and tablets. They utilize the student form to input relevant data and leverage the website's search tool to navigate and access desired information.
* **Frontend Server:** Admission Compass utilizes Vite as its local server for its rapid compilation of frontend code, facilitating efficient development. This server serves frontend files over HTTP to the web browser. Our frontend code, composed of HTML, CSS, and TypeScript, manages requests to and from users. For instance, when students submit data via the website, our server receives it and transmits it to the backend. Similarly, school or degree information is retrieved from the backend and presented to users. APIs are established in both backend servers, enabling the frontend access or calls, such as calling http://localhost:5000/shap from the ML server or http://localhost:8800/results from the database server. Subsequently, the frontend server reformats this data and visually presents it on the site.
* **Database Server:** Our local backend server is built on Node.js and powered by Express.js. This backend serves as the primary handler for HTTP requests from clients and returns responses accordingly. It communicates with the MySQL database to execute database operations such as insertion and selection. The backend processes data received from the frontend, like student submission values and users’ search queries, and forwards them to the database. Additionally, it retrieves dynamic data from the database related to specific schools and degrees, and structures it into arrays for presentation on the frontend.
* **Machine Learning Server:** Admission Compass utilizes a local Python backend server, built on Flask. It handles tasks such as generating machine learning models, making predictions, computing weighted predictions, and calculating SHAP values. The server listens for incoming HTTP requests from the frontend and responds accordingly.
* **Database:** The MySQL database serves as the repository for data utilized by the website. It stores and retrieves various types of information used in Admission Compass. The Node.js backend communicates with the database to execute operations like querying and updating data. It houses tables containing student details, academic records, as well as information about schools and degrees. This database functions as the persistent storage for the website's data, ensuring accessibility and reliability.
* **Admin:** The admin oversees website content and enacts structural adjustments to both the site and the database, as required. Administrators have the authority to export data from the database at regular intervals to review and validate its relevance. Subsequently, they transmit this data to the machine learning server for the purpose of training new models and making predictions, ensuring that the system remains updated and effective.

**Components:**

* 1. **Frontend:**
* **HTML, CSS, and TypeScript:** These files are used to create the user interface.
* **React:** Admission Compass is built with this framework for building a dynamic and interactive site.
* **Vite:** Vite configuration files are used for building and serving our frontend assets during development.
  1. **Machine Learning Server:**
* **Flask**: A micro web framework for Python used to handle HTTP requests and responses.
* **Pandas**: A data manipulation library used for reading and processing CSV data.
* **Scikit-learn**: A machine learning library used for model training and prediction.
* **SHAP**: A library for interpreting machine learning models, particularly useful for feature importance analysis.
* **Matplotlib**: A plotting library used for data visualization.
* **Pickle**: Python's serialization library used to save and load trained machine learning models.
  1. **Database Server:**
* **Express.js**: A web application framework for Node.js used to build RESTful APIs for handling HTTP requests.
* **MySQL2**: A MySQL client for Node.js used to interact with a MySQL database.
  1. **Database Files:**
* **CSV File:** Data files (like student\_data\_final\_testing.csv) which are used for training machine learning models and performing analysis.

**6. Literature Review**

**Background:** Navigating the landscape of higher education can be a formidable task. Elevated admission standards coupled with a myriad of admission avenues contribute to this complexity. Universities are progressively expanding their admission criteria to embrace a wider spectrum of prospective students. Nevertheless, a significant portion of students remains uninformed about the array of alternative admission pathways that exist, potentially hindering their pursuit of higher education opportunities.

**Key Insights:**

* **Higher Education Trends:** Recent trends in higher education indicate a notable increase in enrollment, driven by evolving cultural values and strategic marketing initiatives by educational institutions. Moreover, there is a growing trend among students to apply to multiple institutions simultaneously, a strategy aimed at broadening their options and enhancing their chances of admission.1,2
* **Standardized Test Trends:** In terms of standardized testing, there is a discernible shift away from traditional reliance on standardized tests alone. Institutions are now adopting a holistic approach by considering a diverse range of criteria for admission decisions. This includes not only standardized test scores but also high school performance, completion of preparatory courses, and academic achievements. By incorporating these multiple factors, institutions aim to better predict the academic success of admitted students and ensure a more comprehensive evaluation process.3
* **Predictive Modeling for College Admission:** Techniques like machine learning are common nowadays, in efforts to help students better predict their chances for acceptance. Machine learning algorithms, like XGBoost, LightGBM, and GBM, are used to predict admission. Additionally, exposing influential factors in college acceptances can provide for better understanding and appreciation of each school and degree. 4

1. **Competitive Analysis**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **לימודים בישראל** | **College Scoreboard** | **CollegeData.Com** | **Admission Compass** |
| Prediction tool | Checkmark with solid fill | Close with solid fill | Checkmark with solid fill | Checkmark with solid fill |
| Easy to use | Close with solid fill | Checkmark with solid fill | Close with solid fill | Checkmark with solid fill |
| extensive data input | Close with solid fill | Close with solid fill | Checkmark with solid fill | Checkmark with solid fill |
| No Account creation | Checkmark with solid fill | Checkmark with solid fill | Close with solid fill | Checkmark with solid fill |
| Israel Specific Schools/ Criteria | Checkmark with solid fill | Close with solid fill | Close with solid fill | Checkmark with solid fill |
| Showcases Important features for acceptance | Close with solid fill | Close with solid fill | Close with solid fill | Checkmark with solid fill |

**Competitors:**

1. **לימודים בישראל (Studies in Israel):**
   * **Strengths:** Comprehensive information, prediction tool for admission likelihood.
   * **Weaknesses:** Limited data inputs, confusing layout.
2. **College Scoreboard:**
   * **Strengths:** Clear presentation, streamlined search process, visual data comparison.
   * **Weaknesses:** Limited personal data inputs, no predictive tools.
3. **CollegeData.Com:**
   * **Strengths:** Detailed information, robust admission calculator.
   * **Weaknesses:** Overwhelming amount of information, unclear visualization, requires account creation.

**Website Strategy:**

* Focus on admission requirements for Israeli institutions.
* Collect thorough admission data from accepted college students.
* Implement a user-friendly search tool.
* Integrate an admissions prediction tool with an extensive number of features.
* Compute the important features for acceptance for each school and degree.

**8. Metrics**

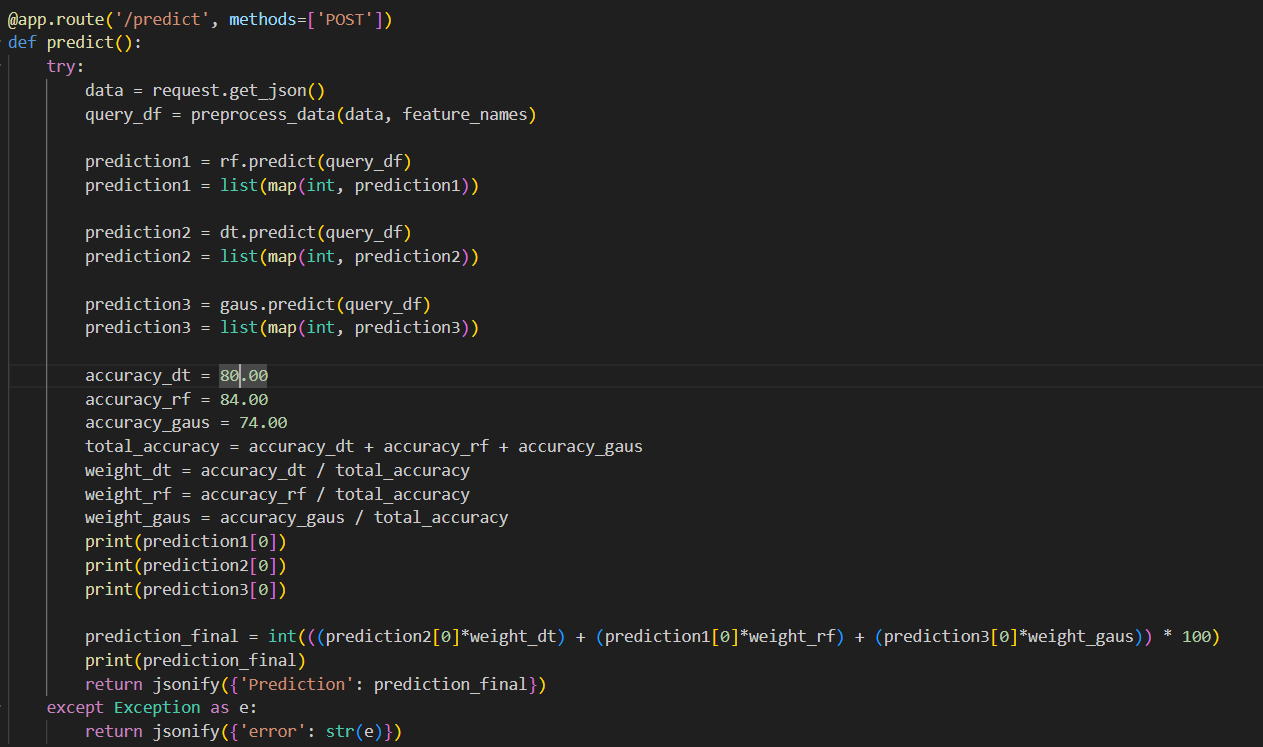
**1. Accepted Student Submissions:** Track the number of submissions to gauge user engagement.

**2. User Feedback:** Gather feedback through surveys and reviews to assess user satisfaction and website success.

**9. Algorithms**

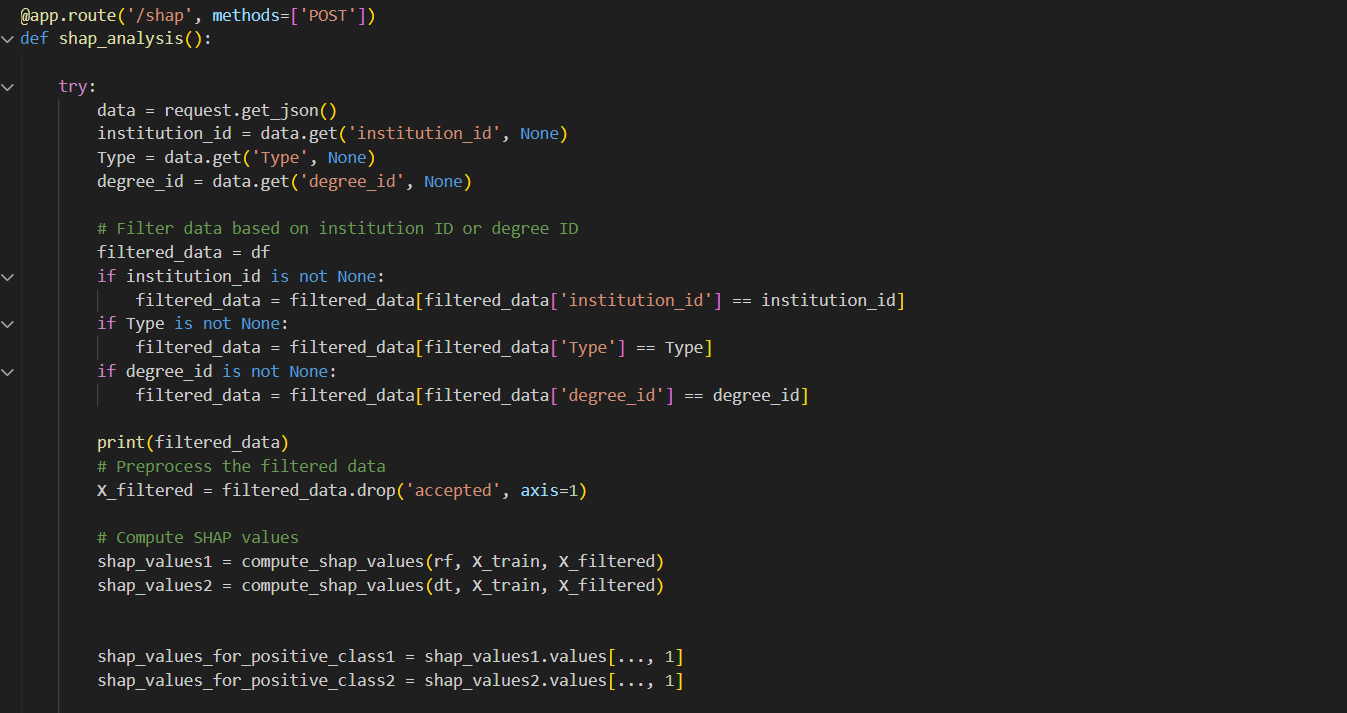
**Predicting Acceptance Algorithm:**

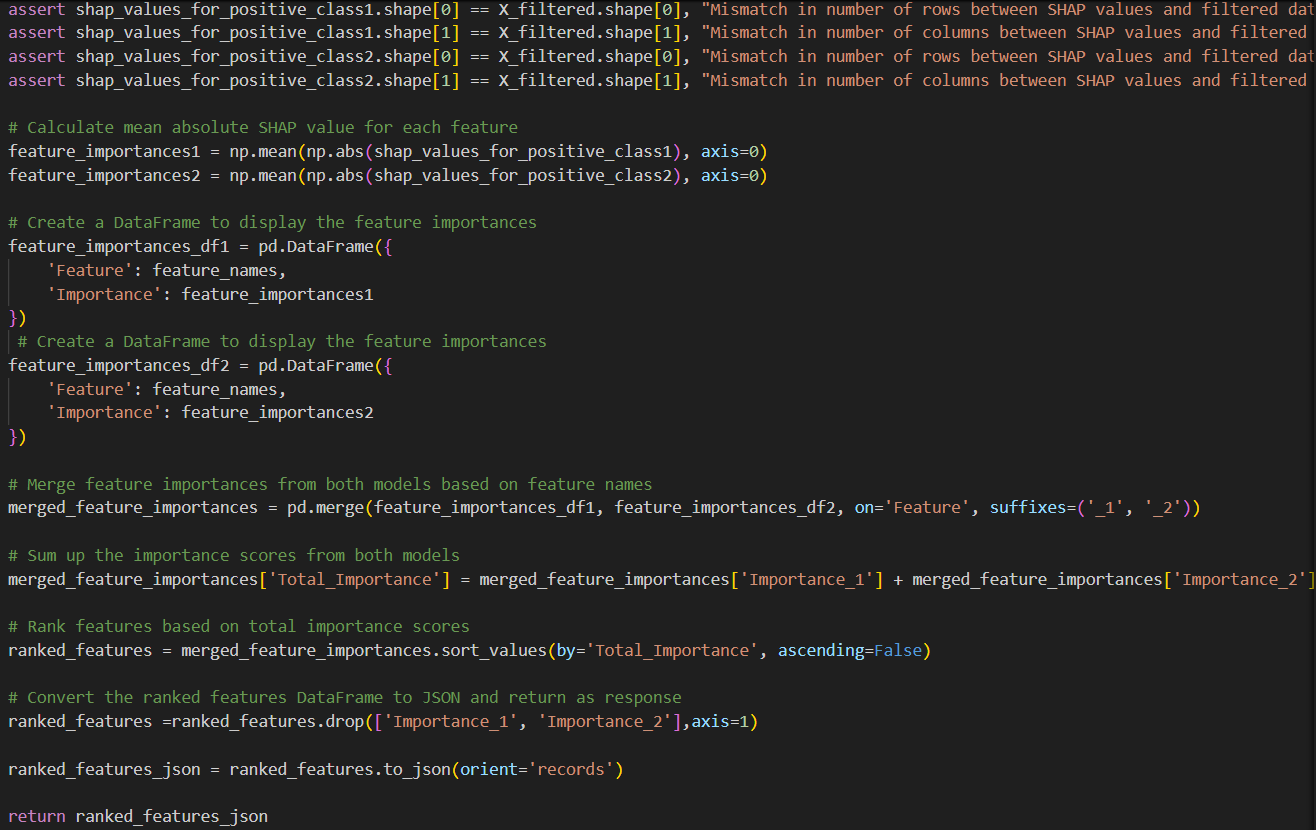
* **Description:** This algorithm predicts the probability of acceptance into a program based on a student's grades.
* **Process:** We use student-submitted data obtained through our website and stored in the database as our dataset. After cleaning the data by removing the predicted/accepted feature, mapping string values in features with numeric values, and filling cells without data with zeros, we proceed to train the data using Decision Trees, Random Forest, and Gaussian Naive Bayes algorithms.
  + **Decision Trees:** This partitions the feature space into a set of disjoint regions and predicts the target variable based on the majority class or average value within each region.
  + **Random Forest:** This uses Decision trees, and each tree is then trained on a random subset of the training data and a random subset of the features. Then it averages the predictions of all the individual trees.
  + **Gaussian Naive Bayes:** This calculates the likelihood of each class given the observed feature values and selects the class with the highest probability as the prediction.

Then, we predict the probability with the new values received from the user using these three models. We then employ weights to determine the joint probability of acceptance. The weights for each prediction are computed by dividing the model's accuracy performance by the total accuracy from each model. Specifically, Random Forest achieved an accuracy rating of 84%, Decision Trees had an accuracy rating of 80%, and Gaussian Naive Bayes attained an accuracy rating of 74%. Using a weighted prediction with multiple models can enhance our prediction accuracy, robustness, and flexibility, making it a more reliable prediction. Finally, we calculate the probability sum, which falls within a range of 0% - 100%, and send it to the user.

**Academic Feature Importance Algorithm:**

* **Description:** Identifies the most influential academic features in the admissions process for a school or degree program.
* **Process:** Firstly, we filter our dataset based on the user's searched school, degree, and degree type, while excluding the target feature. Then, we compute SHAP values using our Decision Tree and Random Forest models, as these models are compatible with SHAP analysis. SHAP (SHapley Additive exPlanations) values leverage game theory to assess the contribution of each feature to a model's prediction for a specific data instance.

After obtaining SHAP values for both models, we isolate the values associated with acceptance into a school (not rejection) and compute the mean values for each feature. Then, employing a voting system, we aggregate the results from both models. Finally, we present these features to the user in a graph, showcasing the top 5 features in descending order of importance.



1. **User Interface Design**

**Home page:** Welcomes users to the website and explains in small detail about the site.

**Student Upload Page:** Allows users to submit their academic details to the site.

**Search Tool Page:** Allows users to search for admission requirements based on various criteria and see available results.

**Detailed Results Page:** Provides details about possible admission requirements and important acceptance features.

**Search By Grades:** Allows users to gauge their prediction probabilities into various schools and degrees.

**About Page:** Explains in further detail the website and our goals.

**11. Database Design**

Database Schema: The following tables outline the structure of our database.

* **institution:** Contains information about institutions, such as name, location, type, and general information.
* **degrees:** Stores details about degrees offered by institutions, including name, type, type id, and associated institution.
* **bagruyot:** Contains Bagruyot (high school matriculation exam) scores and units for various subjects per student.
* **Psychometric scores:** Stores psychometric exam scores, including general grade, verbal score, English score, and quantitative score per student.
* **Student submissions:** Holds submissions from students, including their email, degree choices, institution acceptance/rejection, Bagruyot ID, psychometric id, academic course id, preparatory id, and degree id.
* **preparatory courses:** Stores preparatory scores, including math, physics, English, and scientific writing.
* **student\_preparatory\_courses:** Links students with the preparatory courses they have taken.
* **Electives:** Stores general information about bagruyot electives, including name and id.
* **Academic courses:** Contains information about academic courses, including their names and id.
* **Degree holder:** Stores information about students’ degrees, including number of credits and degree average.

A screenshot of a computer

Description automatically generated

**12. Use Cases**

**Use Case 1: Exploring Admission Requirements**

* **Actor:** Prospective Student
* **Goal:** To explore admission requirements for universities and degree programs in Israel.
* **Steps:**
  1. The prospective student searches for admission requirements using the Admission Compass website.
  2. The system presents a list of search results based on the input criteria.
  3. The student selects a specific program from the search results.
  4. The website provides detailed information about the selected program.
  5. The student can stop here or continue their search in the search tool on the previous page.

**Use Case 2: Previously Accepted Student Submission**

* **Actor:** Accepted Student
* **Goal:** To submit admission criteria and other relevant information to the Admission Compass website.
* **Steps:**
  1. The accepted student fills out a submission form with their admission criteria and other details.
  2. The validated information is stored in the database.
  3. The system displays a confirmation message to the student indicating successful submission.
  4. The system validates the entered information to ensure completeness and accuracy before being added to the dataset for training.

**13. Conclusion**

Admission Compass aims to empower students by offering transparent and easily accessible information regarding admission requirements for various schools and degree programs. Our primary goal is to instill confidence in students by providing them with insights into their acceptance possibilities. Looking ahead, we envision expanding our platform with additional features such as tracking functionality to monitor user searches, an email verification tool to validate student submissions, and enhanced search filters for quicker access to information. Through the integration of advanced search tools, predictive models, and intuitive design, we hope to alleviate some of the stress associated with the college admissions process.

**References**

1. Alon, S., & Tienda, M. (2007). Diversity, opportunity, and the shifting meritocracy in higher education. American Sociological Review, 72(4), 487-511.<https://doi.org/10.1177/000312240707200401>
2. Bridgeman, B., McCamley-Jenkins, L., & Ervin, N. (2000). Predictions of freshman grade-point average from the revised and recentered SAT® I: Reasoning Test. ETS Research Report Series, 2000, i-16.<https://doi.org/10.1002/j.2333-8504.2000.tb01824.x>
3. McDonough, P. M. (1994). Buying and selling higher education: The social construction of the college applicant. The Journal of Higher Education, 65(4), 427–446.<https://doi.org/10.2307/2943854>
4. Protikuzzaman, M., Kanti, M., Kumar, M., & Chandra, B. (2020). Predicting undergraduate admission: A case study in Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Bangladesh. International Journal of Advanced Computer Science and Applications/International Journal of Advanced Computer Science & Applications, 11(12). https://doi.org/10.14569/ijacsa.2020.0111217