Predicting College Admission Results with Machine Learning on Unstructured Online Data

John Tian*
Mira Costa High School
Manhattan Beach, California, USA
john.tian31@gmail.com

Yourui Shao*
BASIS Independent Silicon Valley
San Jose, California, USA
youruishao115022@gmail.com

***Abstract*—College admissions in the United States is a complex and often opaque process, leading to uncertainty and potential unfair biases. This paper presents a novel approach to predicting college admission results using machine learning on unstructured online data. Specifically, we utilize GPT-4o to extract and structure student application details and admission outcomes from more than 4,000 posts on the r/collegeresults subreddit, demonstrating the capabilities of advanced language models in preprocessing unstructured data for machine learning tasks. We employ two distinct methods for predicting admissions results: the first combines descriptive scalars extracted by GPT-4o from unstructured text data with XGBoost and a neural network to predict the probability of acceptance into an institution selectivity tier. The second predicts admission outcomes for specific institutions and compares the effectiveness of tokenization versus descriptive scalars extracted by GPT-4o in representing text features. The models achieve promising results, with Method 1 attaining an accuracy of 91.66% and an AUC-ROC of 0.9298. The results from Method 2 also demonstrate the greater effectiveness of using tokenization (85.1±.2% accuracy) over the explicitly defined features we specified (84.3±.2% accuracy).**

***Index Terms*—college admissions, machine learning, natural language processing, GPT-4o, feature extraction**

I. Introduction

College admissions in the United States is a highly selective process that considers numerous factors, including academic performance, standardized test scores, extracurricular activities, personal essays, and letters of recommendation [1]. The stakes are high: graduates from top-tier universities often have significant advantages in their careers and earning potential.

The admissions process varies significantly among institutions. For example, Harvard University admitted only 3.7% of applicants for the Class of 2028 [2], while the University of California, Irvine had an acceptance rate of 28.8% for the class entering in 2028 [3].

Furthermore, the process lacks transparency, leading to uncertainty and potential biases. A New York Times analysis revealed that at several elite colleges, including five Ivy League schools, more students came from the top 1% of the income scale than from the entire bottom 60% [4]. For applicants with the same SAT or ACT score, children from families in the top 1% were 34% more likely to be admitted than the average applicant, and for those from the top 0.1%, more than twice as likely.

*John Tian and Yourui Shao are co-first authors.

This potential for bias necessitates further data modeling on the admissions process. Our research aims to address this need by developing a machine learning model that can predict college admission results. We utilize GPT-4o, a state-of-the-art language model [5], to extract and structure relevant information from college admission results posts on the Reddit forum r/collegeresults. This novel approach allows us to leverage a rich source of real-world data that reflects a variety of applicants.

To contribute to greater transparency in the college admissions process and provide students with data-driven insights, we have made the model developed in Method 1 (Section III-F publicly accessible at acceptifyai.com. This web-based platform allows users to input their academic and extracurricular information to receive predictions about the tier of colleges they might be admitted to based on the model's analysis.

The key contributions of this paper are:

- A novel method for structuring unstructured online data using GPT-4o.
- Two model approaches for predicting college admission results—one with outputs based on college tiers, the other on specific institutions.
- A comparison of the performance of models trained on features processed using traditional text vectorization versus descriptive scalars extracted by GPT-4o, with the former yielding better results for predicting admissions into specific institutions.
- A publicly accessible tool that provides data-driven insights into the college admissions process.
- Analysis of the most influential factors in college admissions according to the presented models.

By using machine learning to model college admissions results, our work aims to increase transparency in the process, help applicants make more informed decisions, and potentially identify areas where bias may influence admissions decisions.

II. Related Works

Previous studies on machine learning for college admissions prediction have utilized historical admissions data to help admissions officers evaluate future applications.

Before the 2020-21 admissions cycle, admissions bodies often used the submitted standardized test scores on the ACT or SAT to triage applications before human review. However,

when standardized test scores were no longer required as an application component during the COVID-19 pandemic, Lee et al. developed a model to triage applications with additional admissions factors such as class rank and intended majors [6]. Their model grouped applications into ten divisions based on the applications' probabilities of acceptance according to the standardized testing triaging method. Incorporating additional factors besides standardized test scores increased the likelihood of admission from the top pool of the standardized testing method from 82.5% to 91.9%, indicating greater precision in the new triaging method. Additionally, the new top pool was comprised of more underrepresented minority and female applicants, reducing bias from the original standardized testing method.

Using 4442 applications to the Computer Science Department at the University of California, Irvine, which has an admissions rate under 20%, Neda and Gago-Masague use a variety of classifier models such as Random Forest and Multi-layer Perceptrons to predict an application's result [7]. The TextBlob library was used to produce sentiment analysis results and writing-level scores for application essays. The highest accuracy achieved was 78.3% using a logistic regression model. They suggested that the use of a machine learning model reduces human biases, as machine biases can be mitigated with preprocessing techniques.

While these previous works have made significant strides in applying machine learning to college admissions prediction, our approach differs in several key aspects. First, rather than training our model on admissions data from a single institution, we utilize a large, diverse dataset from public online forums which provides a broader perspective on applicant profiles across multiple institutions. To facilitate accurate prediction across multiple institutions, our approach uses both tier-based and institution-specific target values for predictions. Second, our use of GPT-4o for feature extraction from unstructured text data represents a novel application of advanced language models in this domain. Whereas Lee et al. used unigram and bigram tokenization to extract TF-IDF features from text describing extracurriculars and awards, we used and compared a BERT-based tokenizer and GPT-4o for the extraction of such features.

III. Methods

Our approach to predicting college admission results consists of three main components: data collection, feature extraction using GPT-4o, and the development of models based on two distinctive output representations.

A. Data Collection

We collected application data from r/collegeresults, a text-based forum on Reddit where students share posts detailing their college application details and admissions outcomes. Users voluntarily provide their demographic, academic, standardized testing, and extracurricular information; as well as ratings of their application essays and letters of recommendation. The latter metrics were not used in our approach due to inconsistent reporting. In these posts, users also include all colleges to which they applied and the admissions outcome for each college (accepted, rejected, waitlisted, etc.). Additionally, each post is assigned a flair for ease of filtering, which provides information on GPA, SAT/ACT score, field of study, and international student status. See Figure 1 for a post example. Post data was gathered from two sources:

- 961 recent (Jan. 1, 2024, to Jul. 22, 2024) posts (961) were fetched using the Python Reddit API Wrapper (PRAW), limited by Reddit API restrictions.
- 3166 historical (May 6, 2020 to Dec. 31, 2023) posts were obtained from a r/pushshift torrent containing data from the top 40,000 subreddits. These files were in zstandard compressed ndjson format [8].

In total, we collected 4,127 posts from May 6, 2020 to July 22nd, 2024.

Fig. 1. Truncated example of a college results post from r/collegeresults, showcasing the typical format and information shared by applicants, including (but not limited to) demographics, academic stats, and intended major [9].

B. Feature Extraction Using GPT-4o

In this study, we leveraged GPT-4o and GPT-4o-mini (collectively referred to as "GPT-4o") to parse and extract features from the unstructured text data of college application posts. Post bodies were collected as plain text, and GPT-4o was used to transform these unstructured data into a structured format suitable for machine learning models.

Each prompt submitted to GPT-4o consisted of the post flair, body text, a target JSON schema, and a list of key instructions such as skipping posts with uninferrable missing information. We enabled GPT-4o's JSON mode to ensure a valid JSON output. The outputs were a vector of 37 scalars, where each scalar was either categorical (e.g., gender) or ordinal (e.g., number of AP/IB courses taken); and, in a separate call, 16 unprocessed string features, which would undergo further feature extraction methods described in Section III-G (Table I).

The JSON schema specified in the prompt includes detailed instructions on representing post data as scalars that reflect aspects of an applicant's profile. The inclusion of examples for each class and instructions for edge cases in the prompt

ensured not only consistent categorization across diverse posts, but also across multiple GPT-4o calls on the same post (to validate prompt engineering) (See Figure 2).

```
"ethnicity":
  "Integer: 0 = Underrepresented
  Minority in College (Black, Hispanic,
  Native American, Pacific Islander),
  1 = Not Underrepresented Minority
  in College (White, Asian, Other).
  If multiple ethnicities are
  listed, use the one that would be
  considered underrepresented if
  applicable."
```

Fig. 2. Example of a JSON schema specification for the *Ethnicity* class.

C. Data Challenges and Preprocessing

We encountered several challenges in obtaining accurate and representative results with our dataset:

1) Inaccurate or Irrelevant Content: User-submitted posts may contain intentionally falsified information or irrelevant content (e.g., forum announcements). To address this, we instructed GPT-4o to flag posts with insufficient information or evidently false content by returning `{"skip": true}`. For posts missing information for extraction, GPT-4o used contextual inference. For instance, if, in a post, gender was not explicitly stated but "Boy Scouts" was mentioned as an extracurricular activity, GPT-4o would infer and output `0` ("male") for gender. Despite our instructions, GPT-4o did not flag any posts as false, even when presented with posts we suspected might be inaccurate. Out of 4227 collected and parsed posts, GPT-4o identified 138 as lacking sufficient information, resulting in 4089 usable data points.

2) Self-selection Bias and Data Augmentation: The r/collegeresults subreddit predominantly features applicants interested in highly selective institutions, leading to an over-representation of competitive applicants with higher GPAs and more optimal profiles. We used data augmentation to mitigate this bias and enable more accurate predictions for a broader range of applicants.

We generated and added 100 artificially-created "below average" data points with sub-optimal application statistics. These synthetic applicant profiles were created with the following characteristics:

1) The academic indicators *GPA*, *AP/IB Courses*, *AP/IB Scores*, and *Test Scores* were assigned values using weighted random choices or Gaussian distributions biased towards lower values to represent "below average" applicants.
2) All features in the *Extracurriculars* and *Awards* categories were assigned values of 0. Empty strings were assigned for the text features in the *Unparsed* categories. This extreme approach creates a stronger contrast in the dataset, effectively counterbalancing the overrepresentation of applicants heavily involved in these categories in the original data.

TABLE I
LIST OF FEATURES EXTRACTED BY GPT-4O AND GPT-4O-MINI FOR A SINGLE POST, WITH USAGE IN METHOD 1 AND METHOD 2 INDICATED.

Category	Feature	Type	M1	M2
Basic Information	Ethnicity	Categorical (0-1)	✓	
	Gender	Categorical (0-2)	✓	
	Income Bracket	Ordinal (0-4)	✓	✓
	Type of School	Categorical (0-4)	✓	
	Application Round	Categorical (0-1)	✓	
	GPA	Ordinal (0-4)	✓	✓
	AP/IB Courses	Ordinal	✓	✓
	AP/IB Scores	Ordinal (0-4)	✓	✓
	Test Score (SAT/ACT)	Ordinal (0-4)	✓	✓
	Location	Categorical (0-2)	✓	✓
	State Status	Categorical (0-1)	✓	
	Legacy	Categorical (0-1)	✓	
	Intended Major	Ordinal (1-10)	✓	
	Intended Major	String		✓
	Major Alignment	Ordinal (1-5)	✓	
	First Generation	Categorical (0-1)	✓	✓
	Languages	Ordinal	✓	
	Special Talents	Categorical (0-4)	✓	
	Hooks	Ordinal	✓	
	Selectivity of Most Selective Admitted School (Target for Method 1)	Ordinal (0-3)	✓	
Extracurriculars	National/International Activities	Ordinal	✓	✓*
	Regional Activities	Ordinal	✓	✓*
	Local Activities	Ordinal	✓	✓*
	Volunteering	Ordinal	✓	✓*
	Entrepreneurship	Ordinal	✓	✓*
	Internships	Ordinal	✓	✓*
	Additional Academic Programs	Ordinal	✓	✓*
	Research	Ordinal	✓	✓*
	Sports	Ordinal	✓	✓*
	Work Experience	Ordinal	✓	✓*
	Leadership Positions	Ordinal	✓	✓*
	Community Impact	Ordinal (0-4)	✓	✓*
	Average Years of Involvement	Ordinal	✓	✓*
Awards	International Awards	Ordinal	✓	✓*
	National Awards	Ordinal	✓	✓*
	State/Regional Awards	Ordinal	✓	✓*
	Local Awards	Ordinal	✓	✓*
	Other Awards	Ordinal	✓	✓*
Unparsed Extracurriculars	Extracurricular 1	String		✓**
	Extracurricular 2	String		✓**
	Extracurricular 3	String		✓**
	...	...		...
	Extracurricular 9	String		✓**
	Extracurricular 10	String		✓**
Unparsed Awards	Award 1	String		✓**
	Award 2	String		✓**
	Award 3	String		✓**
	Award 4	String		✓**
	Award 5	String		✓**

M1: Method 1, M2: Method 2
* Used in IS-D and IS-T-D of Method 2
** Used in IS-T and IS-T-D of Method 2

3) All synthetic data points were assigned 3 for *Selectivity of Most Selective School*, representing having received admission to only the least selective institutions. This assignment was based on the assumption that applicants with sub-optimal profiles are more likely to be admitted to less selective schools.
4) Other information such as *Ethnicity*, *Gender*, *Income Bracket*, and *Type of School* were randomly assigned to ensure they were not accidentally learned to be correlated with poor admissions results.

The synthetic data generation process aimed to introduce diversity while maintaining realistic relationships between features. For instance, lower GPAs were associated with fewer AP/IB courses taken and lower test scores. Figure 3 visualizes the distributional shifts across key features after incorporating these artificial data points, demonstrating a more balanced representation of applicant profiles.

3) Demographic Skew: In our dataset, 50.4% of the datapoints were assigned a gender of "male," 32.6% "female," and 14.6% "nonbinary, other, or unspecified." These ratios are unreflective of the general applicant pool. For comparison, a 2020 survey from the Association of American Universities (AAU) found that only 1.7% of undergraduate and graduate students identified their gender as transgender, nonbinary, or questioning [10]. The high percentage of "nonbinary, other, or unspecified" in our dataset likely results from the demographic characteristics of Reddit users and instances where users didn't provide sufficient information about gender, leading GPT-4o to classify the gender for those posts as "other." Other features also have similar imbalances. To address these imbalances, we employed the Synthetic Minority Oversampling Technique (SMOTE), as illustrated in Figure 4.

SMOTE is an oversampling technique designed to create synthetic samples for the minority class in imbalanced datasets [11]. The algorithm works as follows:

1) For each sample in the minority class, SMOTE finds its k-nearest neighbors via the Euclidean distance calculation (k=5).
2) One of these neighbors is randomly selected.
3) A new sample at a random point along the line segment joining the original sample and the selected neighbor is created.
4) This process is repeated until the desired balance is achieved.

4) Principal Component Analysis: To gain insight into the underlying structure of our data and potentially reduce its dimensionality, Principal Component Analysis (PCA) was employed. We applied PCA to the dataset used in Method 1 with the implementation provided by the scikit-learn library [13].

Before conducting PCA, we first isolated the ordinal features in our dataset, excluding categorical variables. We then standardized these ordinal features using scikit-learn's StandardScaler [13], which performs z-score normalization. This standardization process ensures that each feature contributes equally to the analysis, regardless of its original scale.

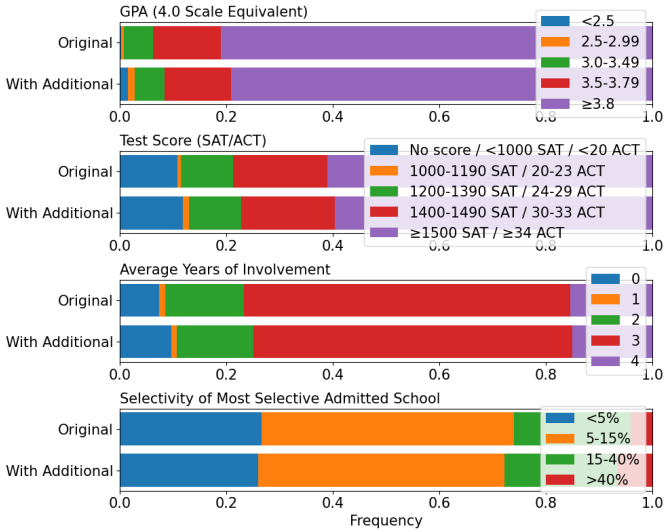

Fig. 3. A comparison of the class frequency distributions for *GPA*, *Test Score*, *Average Years of Involvement*, and *Selectivity* between the untreated post dataset with the dataset with 100 additional artificial data points. Small, but noticeable shifts towards lower values in these four classes can be observed.

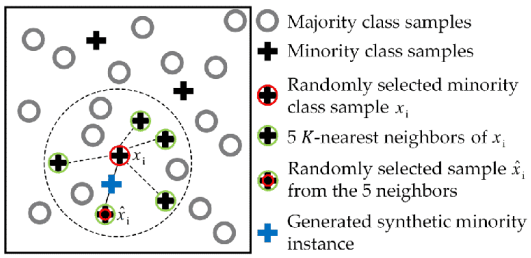

Fig. 4. Illustration of the Synthetic Minority Over-sampling Technique (SMOTE). This method generates synthetic samples of the minority class to address class imbalance, improving model performance on underrepresented categories [12].

Figure 5 shows the scatter plot of the first two principal components. Each point represents an applicant, colored by their acceptance rate category. The plot reveals some interesting patterns:

1) There is a gradient from yellow (highest acceptance rate category, least selective) to purple (lowest acceptance rate category, most selective) moving from left to right, indicating that the first principal component is strongly associated with selectivity.
2) The yellow points (highest acceptance rate category) exhibit the tightest clustering, predominantly in the negative quadrant of both the first and second principal components. This clear separation demonstrates the effectiveness of our synthetic data generation process (as described in Section III-C2).
3) Significant overlap between different categories, especially in the center of the plot, indicating that factors beyond those captured in the first two principal components play a role in determining acceptance rates.
4) Increased spread of points for lower acceptance rate

categories, reflecting the greater variability in features from applications admitted to more selective schools.

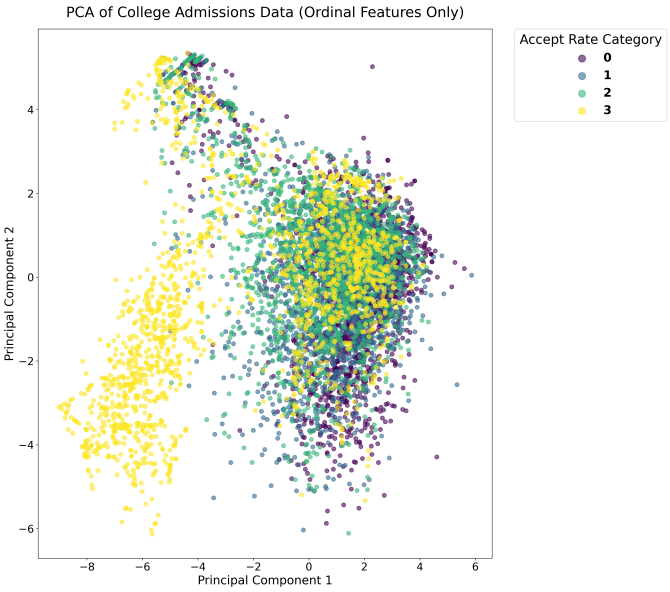

Fig. 5. Principal Component Analysis (PCA) of College Admissions Data.

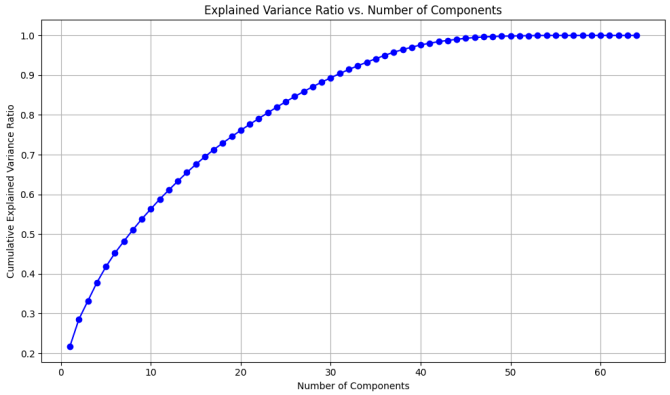

Fig. 6. Cumulative Explained Variance Ratio versus Number of Principal Components.

From Figure 6, we observe:

1) A steep rise for the first 10-15 components, indicating that these capture a large proportion of the variance in the data.
2) The curve begins to level off after about 30 components, suggesting diminishing returns in terms of explained variance for additional components beyond this point.
3) To explain 95% of the variance in the data, 35 components are needed.
4) Even with 60 components, we don't reach 100% explained variance, indicating residual variance from categorical variables not included in the PCA, as well as potential noise in the data.

This PCA analysis suggests that the features extracted by GPT-4o provide non-redundant information about applicants, demonstrating the effectiveness of our feature extraction approach in capturing the multi-dimensional nature of college admissions data. While some dimensionality reduction is possible, the complexity of the data necessitates a relatively large number of features to adequately represent applicant profiles.

D. Feature Engineering

We applied several feature engineering techniques in Method 1 to improve model performance. 9 additional features were added using interaction terms, polynomials, and normalization (see Table II). The 9 categorical features from Table I were also encoded into 17 features using the one-hot encoding technique. This converts each categorical variable into a vector of binary values where the vector dimension corresponding to the category's value is 1, and the remaining dimensions are 0. Adding the 9 engineered features to the original features list and replacing the categorical features with their one-hot encoded counterparts yields 63 total features not including those from *Unparsed Extracurriculars* and *Unparsed Awards*.

TABLE II
ENGINEERED FEATURES, WHERE F_x REPRESENTS THE VALUE OF FEATURE x, AND $x \in X$ REPRESENTS THE SET OF FEATURES IN CATEGORY X. SEE TABLE I FOR THE LIST OF ORIGINAL FEATURES.

Type	**Equation**
Interaction Terms	$F_{\text{GPA}} \times F_{\text{Test Score}}$
	$F_{\text{AP/IB Courses}} \times F_{\text{AP/IB Scores}}$
	$F_{\text{First Generation}} \times F_{\text{Income Bracket}}$
Polynomial Features	F^2_{GPA}
	$F^2_{\text{Test Score (SAT/ACT)}}$
	$F^2_{\text{AP/IB Courses}}$
	$\left(\sum_{x \in \text{Extracurriculars}} F_x\right)^2$
	$\left(\sum_{x \in \text{Awards}} F_x\right)^2$
Normalized Features	$F_{\text{Average Years of Involvement}} / \sum_{x \in \text{Extracurriculars}} F_x$

E. Prediction Approaches

We developed two distinct methods for predicting college admissions results.

The first method seeks to predict the highest tier of college the student will be admitted to, with the target being the *Selectivity of Most Selective Admitted School* feature (see Table I). Then, to predict whether a student will be accepted into a given institution, the model takes into account the selectivity of the given institution, the type of institution it is (e.g., STEM, Liberal Arts), and the competitiveness of the applicant's major at that institution. This works because the model learns the relationship between applicant profiles and admission outcomes across various institution types and selectivity levels during training. During inference, when a user wants to predict admission to a specific school, the model applies this learned relationship to the characteristics of the target school, even if it wasn't explicitly part of the training data. This approach allows the model to generalize

its predictions to any institution, leveraging the patterns it has learned from the diverse set of schools in the training data.

The second method directly predicts the chances of admission into a given school, with the target being a binary reject or accept. Within this method, comparisons between using traditional text vectorization methods and GPT-4o generated features for text representation were also made. We will discuss the procedures for both methods.

F. Method 1: College Tier Prediction

This method aims to predict the selectivity tier of the most selective college to which an applicant will be admitted to. The target output is *Selectivity of Most Selective Admitted School* from Table I, an ordinal variable representing four tiers of college selectivity based on acceptance rates:

- 0: Highly selective (acceptance rate $< 5\%$)
- 1: Very selective (acceptance rate 5-15%)
- 2: Selective (acceptance rate 15-40%)
- 3: Less selective (acceptance rate $> 40\%$)

Let $T = \{0, 1, 2, 3\}$ be the set of selectivity tiers. This method uses an ensemble of two models: XGBoost and a Neural Network. The ensemble prediction represents a scalar in the range $[0, 3]$ which corresponds to the predicted selectivity tier of the most selective college to which an applicant will be admitted. For a given sample, the ensemble prediction is calculated as:

$$\hat{y}_{\text{ensemble}} = \frac{1}{2}\left(f_{\text{XGB}}(\mathbf{x}) + f_{\text{NN}}(\mathbf{x})\right)$$

where:

- $\hat{y}_{\text{ensemble}} \in T$ is the ensemble prediction for a given sample
- $\mathbf{x} \in \mathbb{R}^n$ represents the feature vector of the given sample, where n is the number of features
- $f_{\text{XGB}} : \mathbb{R}^n \to T$ is the XGBoost model function
- $f_{\text{NN}} : \mathbb{R}^n \to T$ is the Neural Network model function

Method 1 uses a logistic function-based approach that takes into account both the ensemble prediction and the selectivity of the school in question. The process can be described by the following equation:

$$P(\text{acceptance}) = \frac{1}{1 + e^{k(\hat{y}_{\text{adjusted}} - x_0)}} \cdot f(\hat{y}_{\text{adjusted}})$$

where:

- $\hat{y}_{\text{adjusted}} = \hat{y}_{\text{ensemble}} - C_{\text{school}}$ is the adjusted prediction, where $C_{\text{school}} \in T$ is the school's selectivity category
- $k = 1$ and $x_0 = 0.5$ are the parameters of the logistic function
- f is an adjustment function defined as:

$$f(\hat{y}_{\text{adjusted}}) = \begin{cases} 1 & \text{if } \hat{y}_{\text{adjusted}} \leq 0 \\ 1 - \min(\hat{y}_{\text{adjusted}}, 0.5) & \text{if } \hat{y}_{\text{adjusted}} > 0 \end{cases}$$

1) XGBoost Model: The XGBoost model was implemented with the following key parameters and optimized with Optuna, a hyperparameter optimization framework [14]:

- Number of estimators: 991, optimized using Optuna
- Maximum tree depth: 10, optimized using Optuna
- Learning rate: $\sim$0.0237, optimized using Optuna
- Subsample ratio of training instances: $\sim$0.7564, optimized using Optuna
- Subsample ratio of columns for each tree: $\sim$0.8624, optimized using Optuna

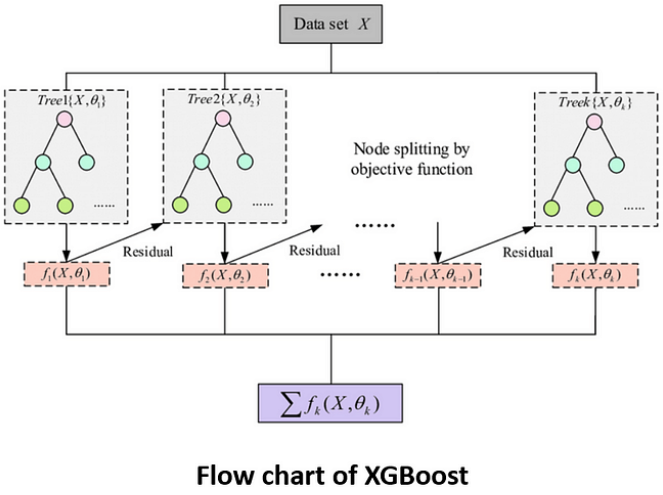

Fig. 7. XGBoost's tree-based ensemble approach, showing how multiple decision trees are combined [15].

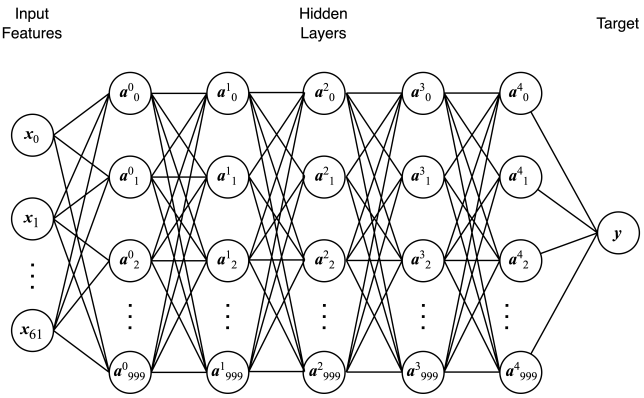

Fig. 8. Neural Network architecture for Method 1, excluding batch normalization and dropout layers.

2) Neural Network Model: The Neural Network model was implemented with the following architecture:

- Input layer: Matches the number of engineered features in length (after flattening)
- 5 hidden layers with 1000 neurons each
- Batch normalization after each hidden layer
- ReLU activation functions
- Dropout layers (rates from 0.4 to 0.1) for regularization
- Output layer: Single neuron with linear activation

The model was compiled with Adam optimizer and Mean Squared Error loss function. Figure 7 illustrates the architec-

ture of the XGBoost. Figure 8 shows the Neural Network model (Logistic Regression) used in Method 1. The complete pipeline for Method 1 is depicted in Figure 9.

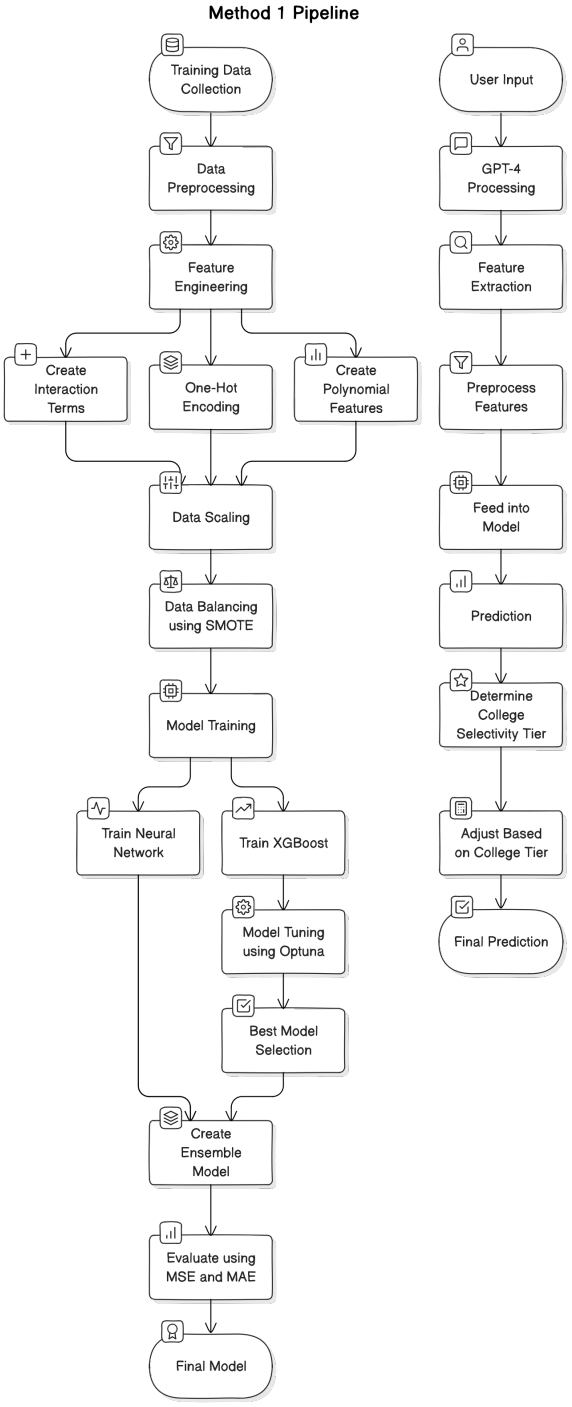

Fig. 9. Method 1 pipeline. The chart on the left illustrates the process for model training, while on the right is the process for inference.

G. Method 2: Institution-Specific Prediction

Method 2 aims to predict admissions outcomes for specific institutions. To achieve this, we expanded our dataset to include multiple admission results per post, as applicants typically apply to several schools. Using GPT-4o, we extracted the unabbreviated school name, admission round, and applicant's in-state status from the raw post text (Table III). This process expanded our initial 4,189 data points (before applying SMOTE, as described in Section III-C2) to 21,970 data points, with each representing a unique school application from a single post. This expansion enables our model to predict admission outcomes for individual institutions within the dataset.

1) Additional Features: Institutional characteristics significantly influence admissions outcomes. Colleges and universities vary not only in overall selectivity but also in the popularity and competitiveness of specific majors within their programs. For example, a particular institution might have a highly selective computer science program but a less competitive humanities department. To accurately predict admissions for individual institutions, our model incorporates these nuanced factors as inputs.

With a 2018 dataset based on the Common Data Set, we obtained the college admission rates and third-quartile SAT evidence-based reading and writing (EBRW) and math score values for admitted students across all institutions. These metrics serve as indicators of selectivity, with lower admission rates and higher test scores corresponding to greater selectivity [16] (Table III).

Additionally, we acquired data on the number of students completing degrees in various areas of study for all institutions from mid-2021 to mid-2022 from the National Center for Education Statistics [17]. We consolidated these areas of study into 12 general categories and calculated frequency distributions across these categories for each school. For each post, we employed keyword mapping to identify which of the 12 categories the text version of the *Intended Major* feature belongs to (Table I). The frequency of that major category and the category itself was then incorporated as additional features for the 21,970 data points (Table III).

TABLE III
ADDITIONAL FEATURES GATHERED FOR THE INSTITUTION-SPECIFIC APPROACH.

Source	Feature	Values
GPT-4o extraction	Admissions Round	Categorical (4): REA/EA, ED, EDII, RD
	In-State Status	Categorical (2): in state, out-of-state/not applicable
External data	SAT EBRW Third Quartile	Numerical
	SAT Math Third Quartile	Numerical
	Admissions Rate	Numerical (%)
	Major Popularity	Numerical (%)
	Major	Categorical (12)

2) Comparing Text Vectorization: In developing Method 2, we sought to compare traditional vectorization techniques for text with the descriptive abstraction technique using GPT-4o. Extracurricular activities and awards are typically represented

as text descriptions in college applications and in our source posts. As described in section III-B, GPT-4o vectorized these text descriptions by categorizing them into types such as "internships" and "regional" and aggregating counts of these types (see *Extracurriculars* and *Awards* in Table I).

We compare this approach with a traditional word vectorization technique: tokenization. Tokenization splits text sequences into tokens from a predefined vocabulary, which can then be converted into a vector of numerical ID values. Table IV provides an example of this process.

For our implementation, we utilized the Base Uncased DistilBERT Tokenizer and Model [18]. The tokenizer converted each extracurricular from *Unparsed Extracurriculars* (Table I) into a vector of length 15, and each award from *Unparsed Awards* into a vector of length 10, yielding 200 token ID features in total.

No further word embedding procedures, such as those using models like DistilBERT [18], were implemented. Through testing beyond the scope of this paper, we discovered that incorporating DistilBERT embeddings as inputs, rather than token IDs, worsened validation performance. This was likely due to the embeddings not being fine-tuned for the downstream task of identifying extracurricular activities and award descriptions that influence college admission probabilities. A potential concern arises from the use of token IDs, which are categorical data represented as integer values, which the decision tree will treat as ordinals. While we could convert each of the 200 token ID features into a large vector using one-hot encoding, thereby transforming the features into treatable binaries; this approach is not preferred as it would significantly increase the number of input dimensions.

So, can the decision trees described in Section III-G3 effectively handle categorical data with a large number of classes that are not one-hot encoded? The results presented in Section IV-A suggest so. It appears that the scikit-learn decision tree implementation (detailed in Section III-G3) can partition the possible values for categorical features in such a way that tree splits can effectively determine whether a feature belongs to individual classes.

TABLE IV
EXAMPLE OF TOKENIZATION OUTPUT WITH A MAXIMUM LENGTH OF 10 TOKENS

Input Text	`"National Merit Semifinalist"`
Tokens	`['[CLS]', 'national', 'merit', 'semifinal', '##ist', '[SEP]']`
ID Vector	`[101, 2120, 7857, 16797, 2923, 102, 0, 0, 0, 0]`

This approach allowed us to compare the effectiveness of GPT-4o's abstraction technique with a more traditional NLP tokenization method in the context of college admissions prediction. Thus, we developed three distinct institution-specific models:

- IS-D (Descriptive): This uses only the 18 descriptive features under *Extracurriculars* and *Awards* from Table I to represent extracurricular activities and achievements.
- IS-T (Token-based): This model uses only the 200 token IDs generated through tokenization to represent extracurricular activities and awards.
- IS-T-D (Hybrid Approach): This model combines both approaches, incorporating the 18 features from Table I and the 200 token IDs to represent extracurricular activities and awards.

3) Model Architecture: As data prior to SMOTE processing was used for this method, demographic features susceptible to imbalance were completely disregarded during training. Other features extracted for the first method were also discarded. Thus, only the following features were used:

- *Income Bracket*, *GPA*, *AP/IB Courses*, *AP/IB Scores*, *Test Score*, *Location*, and *First Generation*.
- Representations for extracurriculars and awards, whether the 18 descriptive features, 200 token IDs, or both (see III-G2).
- The 7 institution-specific features from Table III.

We used the RandomForestRegressor model from the scikit-learn library, which follows a Random Forest model architecture with 1000 decision trees [13]. A maximum of $\log_2 N$ features, where N is the total number of features, are considered at each split in a tree. Our target values for training are binary—0, representing reject, and 1, accept. Using a regressor, the model outputs fall in the range [0,1], thus also representing the confidence of the prediction. During inference, this offers a more satisfactory answer than a binary accept or reject.

TABLE V
MODEL PERFORMANCE METRICS ON VALIDATION SETS. FOR MODELS UNDER METHOD 2, THE 95% CONFIDENCE INTERVAL ON 20 SAMPLES IS PROVIDED (SEE SECTION IV-A2).

	Method 1			**Method 2**		
Model	**XGBoost**	**NN**	**Ensemble**	**IS-D**	**IS-T**	**IS-T-D**
MSE	.0836	.2510	.1204	.111±.001	.110±.001	.110±.001
MAE	.1226	.3833	.2459	.230±.001	.242±.001	.241±.001
Accuracy	.9166	.7261	.8929	.843±.002	.851±.002	.850±.002
Precision	.9198	.7271	.8973	.865±.003	.867±.002	.867±.003
Recall	.9166	.7261	.8929	.878±.003	.890±.003	.888±.002
AUC-ROC	.9298	.7830	.9095	.834±.002	.841±.002	.840±.002

IV. RESULTS

A. Model Training and Evaluation

1) Method 1: Method 1 was trained using k-fold cross-validation with k=5. The training process for Method 1 involved using Optuna for hyperparameter tuning, run on 100 trials. Method 2 was trained and evaluated using an 80-20 train-test split.

As multiple models were ensembled, the accuracy for Method 1 was calculated as:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\left(\lfloor y_i + 0.5\rfloor = \lfloor \hat{y}_{\text{ensemble},i} + 0.5\rfloor\right)$$

where:

- N is the total number of samples
- $y_i \in \{0, 1, 2, 3\}$ is the true label for the i-th sample
- $\hat{y}_{\text{ensemble},i} \in [0, 3]$ is the ensemble prediction for the i-th sample
- $\mathbb{I}$ is the indicator function, which outputs 1 if the condition is true and 0 otherwise
- $\lfloor \cdot + 0.5 \rfloor$ is the mathematical procedure for rounding to the nearest integer

Table V shows the performance metrics on validation sets for both methods.

In Method 1, despite the XGBoost model demonstrating superior performance on the current dataset, we opted for an ensemble approach combining XGBoost and Neural Network predictions. This decision was based on empirical observations suggesting that the ensemble model exhibits improved generalization to real-world applications beyond the potentially biased training set.

2) Method 2 Tokenization: In Method 2, we trained 20 instances of each model (IS-D, IS-T, and IS-T-D). These instances differ on the randomized train-test split and the 1000 randomly generated decision trees. The performance metrics for each of the 20 instances were used to calculate a 95% confidence interval for the mean population value of the given metric across all possible trained instances of a model (see Table V). This allows for a thorough comparison of model performance metrics which accounts for the variation in performance between trained instances of Random Forest models.

We first use an independent two-sample t-test on the mean accuracies (μ_{accuracy}) between IS-T and IS-D to examine whether IS-T achieves a truly greater mean accuracy. The hypotheses are as follows:

$$H_0 : \mu_{\text{accuracy, IS-T}} = \mu_{\text{accuracy, IS-D}}$$
$$H_A : \mu_{\text{accuracy, IS-T}} > \mu_{\text{accuracy, IS-D}}$$

With a degree of freedom (df) of 38 ($2N-2$ where $N = 20$ for the 20 trained instances of each model, or our sample size), we achieved a test statistic $t = 5.922$, which yields a p-value of $3.641 \cdot 10^{-7} < 0.025$. As such, we can reject the null hypothesis and conclude that IS-T outperforms IS-D in accuracy. The same process for MSE, recall, and AUC-ROC also concludes that IS-T yields better true performance metrics, but tests for MAE and precision were inconclusive.

In terms of accuracy, we can conclude that traditional text vectorization techniques generally offer more comprehensive representations of extracurricular activities than the descriptive features extracted with GPT-4o. More optimal descriptive features may be chosen for extraction to improve IS-D performance; however, the traditional text vectorization process could also be optimized by adding college application-specific vocabulary or fine-tuning an NLP model to embed tokens according to linguistic relations.

Similar t-tests on metrics between IS-T and IS-T-D failed to reject the null hypothesis. Thus, there is insufficient evidence that the models differed in performance and that one outperformed the other. One possible explanation for a lack of performance increase with the addition of descriptive features in the IS-T-D model is that the descriptive features did not offer additional information to what was already provided by token IDs. Further work should examine whether engineering better descriptive features for GPT-4o to generate can eventually exceed the interpretive capability of decision trees on token IDs (Section V-B4).

B. Feature Importance Analysis

Feature importance analysis was conducted to understand which factors most significantly influence college admissions predictions. The feature importances were determined through the Gini Gains of each feature across all splits of each model's decision trees, which measures the frequency at which a feature is used in splitting branches. For both Method 1's XGBoost model and Method 2's IS-T, the calculated importance was provided by the `feature_importances_` attributes from the scikit-learn model implementations [13].

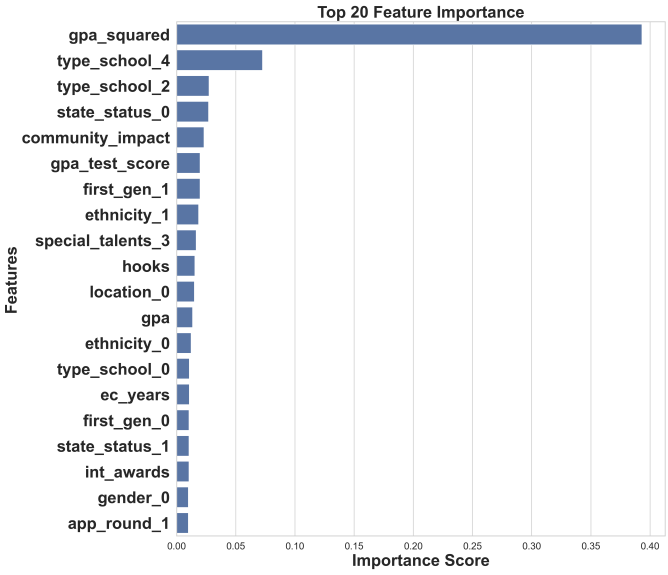

Fig. 10. Top 20 Feature Importance for Method 1 (College Tier Prediction). The graph illustrates the relative importance of various factors in predicting college admission tiers. Notably, GPA-related features dominate the top positions, with school type and community impact also showing significant influence.

1) Method 1: College Tier Prediction: Feature importance analysis was conducted to understand which factors most significantly influence college admissions predictions. Figure 10 shows the top 20 most important features for the XGBoost model from Method 1, while Figure 11 illustrates the impact and direction of influence for these features on the model output.

The top three features with the highest impact on model predictions are:

1) `community_impact` (28.25%)
2) `hooks` (18.28%)
3) `gpa_test_score` (8.26%)

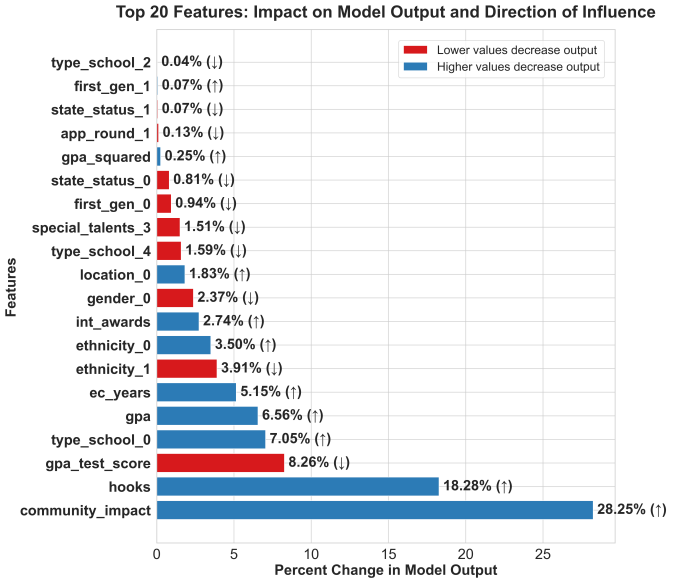

Fig. 11. Top 20 Features: Impact on Model Output and Direction of Influence. This graph shows the percent change in model output for each feature and whether higher or lower values of the feature lead to admission to more selective schools (lower model output).

For categorical variables that have been one-hot encoded, the notation is as follows:

- `feature_name_x`: Represents the binary indicator for a specific category within a categorical feature.
- `x`: An integer representing the category's index.

For example, `gender_0` represents male gender, while `ethnicity_1` represents non-underrepresented minorities. The percentage for these binary features indicates the change in model output when the feature is present (1) versus absent (0).

These percentages represent the maximum potential impact each feature has on the model's output, calculated as follows:

- For continuous features, the data was divided into 10 equal-width bins. For categorical features, each unique category was treated as a separate bin.
- Average model outputs (predicted selectivity tier) were calculated by changing the value of a feature across all data points to the value corresponding to each of the feature's created bins. All other features (features not being analyzed) were held constant at their original values from the dataset.
- The percent change was calculated as follows:

$$\frac{\overline{y}_{\text{bin with highest output}} - \overline{y}_{\text{bin with lowest output}}}{\overline{y}_{\text{bin with lowest output}}} \times 100\%$$

  where bins are 10 equal-width divisions of the feature's range for continuous features and binary indicators for categorical features as they are one-hot encoded.

This method isolates the effect of each feature by examining how the model's predictions change across its range of values while keeping other features unchanged. The percentage indicates the maximum difference in model output attributable to that feature alone.

For example, a feature impact value of $x\%$ for `community_impact` means that, all else being equal, applicants with the highest community impact scores received, on average, tier prediction values $x\%$ lower (more selective) than those with the lowest community impact scores. In this case, $x = 28.25$.

It's important to note that:

- These percentages represent the mean change in outcome and are not indicative of the change in outcome for a specific data point.
- The binning process, which divides continuous features into 10 equal-width bins, allows for systematic comparison across features but may obscure within-bin distribution details. For instance, it doesn't account for potential skewness within bins or the precise location of data points, which could be relevant for features with non-uniform distributions.

The following section provides an interpretation of the top three features and key demographic features:

- `community_impact` (28.25%) quantifies an applicant's community impact on a 0-4 scale. Notably, this highly influential feature was extracted by GPT-4o from unstructured text, demonstrating the effectiveness of AI-driven feature extraction in admissions prediction.
- `hooks` (18.28%) represents the number of significant factors that may provide a notable advantage in the college admissions process such as being a recruited athlete or having a significant hardship (e.g. homelessness).
- `gpa_test_score` (8.26%) is the product of the extracted ordinal values of the applicant's GPA and SAT/ACT score. Counterintuitively, the model correlates lower values to more selective institutions. This may be because feature impact analysis was only done for XGBoost and not the ensemble model, which, based on empirical observations, makes better generalized predictions beyond the training set.
- `ethnicity_1` (3.91%) represents non-underrepresented minorities. The negative percentage change indicates these applicants were associated with less selective admissions compared to underrepresented minorities in our model. However, the dataset used in this analysis partly precedes the U.S. Supreme Court's decision to ban race-conscious admissions policies in June 2023. This would affect the relevance of the interpretation of ethnicity-related features in the model to current admissions outcomes.
- `gender_0` (2.37%) represents male applicants. The negative percentage indicates male applicants in our dataset were associated with slightly less selective admissions compared to applicants who identified as female or other.
- `first_gen_0` (0.94%) represents non first-generation students. Our model associates this with less selective admissions, agreeing with common policies that favor

first-generation students.

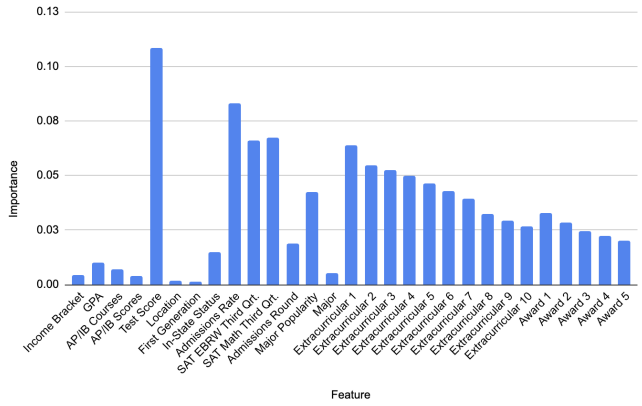

Fig. 12. Feature importances for IS-T from Method 2.

2) Method 2: Institution-Specific Prediction: See Figure 12 for the feature importances of the IS-T model. To measure the importance of each extracurricular and award text feature tokenized as input for IS-T, the feature importance of the corresponding token IDs was summed.

A few hypotheses can be drawn to explain some peculiarities in feature importances:

- The model minimally considers the demographic features *Income Bracket*, *Location*, and *First Generation*. This differs from the analysis of Method 1, where demographics are of importance for the model. This likely demonstrates the importance of SMOTE for accounting for the features that have class imbalances in the dataset.
- An applicant's *Test Score* is the feature of the greatest importance, while other academic factors such as *GPA*, *AP/IB Courses*, and *AP/IB Scores* are minimal in importance. This can indicate that the information provided by other academic factors is redundant to that of *Test Score*; or that the other academic factors are unreliable for prediction. If the latter is true, it may be because GPA calculations vary from high school to high school and the admissions expectation for the number of AP or IB courses taken is dependent on the number available to the applicant.
- Extracurriculars and awards decrease in importance according to the order in which they were in the posts and the applications. This is likely a reflection of the advice commonly given to applicants—to place their extracurriculars and awards in descending order of significance and value.

V. Conclusion

A. Discussion

The feature importance analysis highlights the critical role of academic performance in college admissions predictions, with GPA dominating in Method 1 and standardized test scores in Method 2. This aligns with the general understanding that academic achievement is key in college admissions. However, the significance of extracurriculars, awards, and demographic factors among top features demonstrates the complex nature of the US admissions process.

The PCA results further support this complexity, as a relatively high number of principal components were required to explain 95% of the variance in admissions results. This suggests that attempts to oversimplify the process or rely on just a few factors may miss important nuances.

The clear gradient in the PCA scatter plot suggests that there are indeed distinguishable patterns in the data that correlate with acceptance into tiers of institutions. We inferred that institutions of higher selectivity have more stringent and perhaps more consistent criteria for admission.

B. Limitations

While our models show promising results in predicting college admissions, there are several limitations to consider:

1) Data Source Bias: Our data source, r/collegeresults is not representative of the population of all applicants. While artificial data and SMOTE were used to address this issue, further testing is necessary to conclude whether the models generate accurate predictions for a more representative sample.

2) Limited Contextual Understanding: Although GPT-4o is a powerful language model, it may miss nuanced contextual information in applicant profiles. This could lead to misinterpretations of complex applicant characteristics during data extraction despite good prompt engineering.

3) Missing Admissions Factors: As we do not have access to full student applications, our model does not account for several important factors in the admissions process, such as essays, letters of recommendation, interviews, and demonstrated interest.

4) Missing Features: The process of abstracting text descriptions into descriptive features may lead to information loss. As discussed in Section IV-A, improving vectorization using enhanced word embeddings may yield better results.

C. Future Work

To address these limitations and further improve our models, we propose the following areas for future research:

1) Data Sourcing: To address biases noted in Section III-C2, future work should incorporate more representative datasets. Also, in order to enhance model comprehensiveness and accuracy, these datasets should include critical application components mentioned in Section IV-A, such as essays and recommendation letters.

2) Essay Analysis: With datasets containing critical application components, methods to analyze application essays can be developed. Similar to the approach taken by Neda and Gago-Masague discussed in Section II, advanced NLP techniques for assessing qualitative features such as writing quality, creativity, and personal growth are needed to quantify their impact on admissions decisions.

This study presents a novel approach to predicting college admissions using machine learning techniques on unstructured

online data. Achieving accuracies of 91.66% and 85.1±.2% respectively, our two methods—one for predicting admission to college tiers and another for individual institutions—both show promising results.

ACKNOWLEDGMENTS

We would like to thank Dr. Hugh Tad Blair from the Department of Psychology at the University of California, Los Angeles, for helping us procure funding and giving guidance on this project; And the California State Summer School for Mathematics & Science (COSMOS) for bringing us together, providing us the opportunity to combine our wisdom and efforts, allowing us to meet some of the most intellectually compelling people in our lives.

REFERENCES

[1] National Association for College Admission Counseling, "Factors in the admission decision," 2023, [Online; accessed 8-August-2024]. [Online]. Available: https://www.nacacnet.org/factors-in-the-admission-decision/

[2] Harvard College, "Admissions statistics," 2024, [Online; accessed 8-August-2024]. [Online]. Available: https://college.harvard.edu/admissions/apply/admissions-statistics

[3] IvyWise, "Class of 2028 admission rates," 2024, [Online; accessed 9-August-2024]. [Online]. Available: https://www.ivywise.com/blog/college-admission-rates/

[4] A. Bhatia, C. C. Miller, and J. Katz, "Study of elite college admissions data suggests being very rich is its own qualification," *The New York Times*, July 2023, [Online; accessed 8-August-2024]. [Online]. Available: https://www.nytimes.com/interactive/2023/07/24/upshot/ivy-league-elite-college-admissions.html

[5] OpenAI, "Hello GPT-4o," May 2024, [Online; accessed 10-August-2024]. [Online]. Available: https://openai.com/index/hello-gpt-4o/

[6] H. Lee, R. F. Kizilcec, and T. Joachims, "Evaluating a learned admission-prediction model as a replacement for standardized tests in college admissions," in *Proceedings of the Tenth ACM Conference on Learning@ Scale*, 2023, pp. 195–203.

[7] B. M. Neda and S. Gago-Masague, "Feasibility of machine learning support for holistic review of undergraduate applications," in *2022 International Conference on Applied Artificial Intelligence (ICAPAI)*. IEEE, 2022, pp. 1–6.

[8] Watchful1, "Separate dump files for the top 40k subreddits, through the end of 2023," Reddit post and GitHub repository, Reddit, February 2024, data prior to April 2023 collected by Pushshift, data after that collected by ArthurHeitmann (https://github.com/ArthurHeitmann/arctic_shift). Extracted, split and re-packaged by Watchful1. Hosted on academictorrents.com. [Online; accessed 8-August-2024]. [Online]. Available: https://redd.it/1akrhg3

[9] u/DeresingMoment, "math major gets mogged by private schools," Jul. 2024. [Online]. Available: redd.it/1e3mboj

[10] Postsecondary National Policy Institute, "Lgbtq+ students in higher education," 2023, [Online; accessed 10-August-2024]. [Online]. Available: https://pnpi.org/factsheets/lgbtq-students-in-higher-education/

[11] S. Satpathy, "Smote for imbalanced classification with python," *Analytics Vidhya*, August 2024, [Online; accessed 9-August-2024]. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/

[12] Admin, "The basic principle of the synthetic minority oversample technique (smote) algorithm," 2017, [Online; accessed 8-August-2024]. [Online]. Available: https://rikunert.com/smote_explained

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[14] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," *arXiv preprint arXiv:1907.10902*, 2019.

[15] Eason, "Xgboost or logistic regression model for diabetes prediction," *Medium*, July 2022, [Online; accessed 8-August-2024]. [Online]. Available: https://medium.com/p/1c3670cbbf6e

[16] S. Qian, "College Admissions," November 2018. [Online]. Available: https://www.kaggle.com/datasets/samsonqian/college-admissions

[17] N. C. for Education Statistics, "Completions/awards/degrees conferred by program (cip), award level, race/ethnicity, and gender," 2022. [Online]. Available: nces.ed.gov

[18] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2020. [Online]. Available: https://arxiv.org/abs/1910.01108