International Student Enrollment at a US University:
First Source Analysis and Future Enrollment Predictive Models

Abstract

Using data analysis, research and predictive classification models, this project aims to examine lead sources, identify influential factors for student enrollment among applicants, and determine the importance of third-party name purchases for higher education.

For non-selective institutions with ambitious growth goals, acquiring cold leads or third-party inquiries is vital, as organic engagement is generally not sufficient to meet enrollment goals. Given the rise of domestic and international student data privacy laws, there is a risk that universities may no longer be able to purchase the names of students from third-party suppliers and partners.

With the above challenges in mind, this project also aims to build and compare the performance of various classification models that can be utilized to make predictions about applicants for more robust projections of future enrollment.

Background and Terminology

The United States alone has over 4,000 colleges and universities. The QS World University rankings feature over 1,500 universities across 105 countries worldwide.[1] With growing mobility and economic recovery since the pandemic, many students are considering and pursuing higher education options outside of their home countries. On the side of the universities, there is tremendous competition for recruiting such students, who are often a vital source of revenue and an important asset for their communities.

While some students may say that marketing had little to do with their university choice, the reality is that international student recruitment often involves extensive proactive marketing campaigns. Even the most competitive universities are setting aggressive international enrollment targets, because they realize that many international prospects begin their college search with multiple preferred institutions in mind. They recognize the fierce battle underway to attract the very best talent and thus work hard to build awareness and interest, often targeting prospective students at a very young age. In addition to in-person and virtual engagement, this involves buying contact lists of prospective international applicants and targeted digital advertising that puts universities in the students' radar instead of hoping for organic interest.

Cold leads represent the very top of the college recruitment funnel. For many institutions, most non-inquiring international “cold” leads are generally high school students

[1] https://www.topuniversities.com/world-university-rankings

who have registered for (and/or taken) a TOEFL, IELTS, AP, SAT, or ACT exam. Cold leads can also include high school students who have taken a survey or opted in through a search platform. Many institutions also refer to these students as "*prospects*." Targeted marketing materials and content are sent to *prospects* to introduce the university and cultivate interest. If a student chooses to respond to a marketing push, they are then considered to be an *inquiry*. It is at this point that a university will begin to encourage the student to start and complete an application. The *applicants* are in turn evaluated and become *admits* if they meet admission requirements. Finally, *admits* are encouraged to accept their offer of admission by paying an enrollment deposit, and once a *deposited* student registers for classes and orientation, they are generally considered an *enroll*.

Motivation and Challenges

Internal student data representation

We are one of the largest public universities in the United States with over 100,000 enrolled students in its on-campus and online programs. International students alone account for more than 14,000. Despite (and perhaps also because of) this impressive size, student data is rather fragmented across multiple systems.

The university uses the Salesforce customer relationship management (CRM) platform to store lead and inquiry data, as well as all organic engagement between each student and the university – this includes high school visits, fairs, as well as on-campus and virtual events. Additionally, PeopleSoft is used to manage a student's application, including all application materials like transcripts. As Salesforce is not custom-designed for the higher education context, the different data points available for each student lead can make it challenging to quickly determine the earliest point of contact or the true first source of a student record. "First source" is defined as the channel or third-party source by which the university first learned of a prospective student; applicants who do not have such a first source in the data are referred to as *stealth applicants.*

Rising student privacy laws

Student Personally Identifiable Information (PII) is an essential part of cold lead data. In addition to basic details and contact information, PII includes academic performance indicators, family data (such as income indicators), and other protected data. Institutions use this information to identify new leads by cross-referencing it with their internal CRMs. This data is also used to inform targeted recruitment campaigns.

To mitigate the risks of leaking such sensitive information and prevent misuse, privacy laws have been adopted in the United States over recent years at the state level that are often modeled on Europe's General Data Protection Regulation (GDPR). For the US domestic student market, this has resulted in what is called a "search cliff" or a projected drop in student names available to purchase from the College Board.[2] Internationally, we are witnessing a similar trend: looking at our two largest international student markets, there is China's Personal Information Protection Law (PIPL)[3] passed in 2021 and India's Digital Personal Data Protection Bill passed in 2023.[4] Such laws are also likely to make international cold lead cultivation and gathering more challenging for third-party vendors like the College Board, ETS (creator of TOEFL) and others. This in turn means that universities may be up against a corresponding "search cliff" issue internationally in the years to come.

In addition to looking at the relative importance and performance of our lead sources, it is crucial for us to be able to predict and forecast our international enrollment with higher degrees of precision for leads at different stages within our funnel. To this end, predictive models will be explored with the aim of classifying students based on various data points available in our internal systems.

Datasets

The data leveraged included de-identified and anonymized recruitment data, which allows to determine how the university first obtains the information of its prospective international students. It was composed of multiple reports from two sources: Salesforce and PeopleSoft, spanning four total recruitment cycles: 2021-24.

- Salesforce reports are pre-application data that is broken up into five file groups with 13-20 columns each:
    - ❖ **SF_Leads-Inqs_Base**: One row for each unique lead/inquiry from Salesforce, filtered to International (or records without a listed country).
    - ❖ **SF_Leads**: all cold lead sources for international students (or students with a missing country value). Students can have multiple lead sources (multiple rows).

[2] https://go.collegevine.com/search-cliff-calculator
[3] https://www.aacrao.org/advocacy/compliance/china%27s-personal-information-protection-law-pipl/china%27s-new-privacy-law-u.s.-cui-regulations-spark-confusion
[4] https://prsindia.org/billtrack/digital-personal-data-protection-bill-2023

- ❖ **SF_Other_Source**: all “non-case” inquiry sources including third-party hand-raiser channels. Students can have multiple inquiry sources (multiple rows).
- ❖ **SF_Cases**: all relevant case sources (inbound email, in person appointments, incoming phone calls, webforms). Students can have multiple Salesforce cases (multiple rows).
- ❖ **SF_Events**: all relevant event and visit sources. Students may have more than one event/visit source.

- PeopleSoft data is application data that has been aggregated into separate reports to include our applicant pools for Fall 2021 – Fall 2024. Each record contains 75-78 columns.

Methodology

The first stage involved significant data wrangling and cleaning of the fragmented reports to create a more meaningful and interpretable picture of our international student lead sources. This process was repeated for each year of data from 2021 to 2024 with the help of Python and the pandas library specifically. Summaries and graphs of the resulting lead sources were generated using several tools including Tableau, Excel, and Juilus.ai

The second stage dealt primarily with application data that was merged with first source data obtained in the initial stage. This data was further cleaned to exclude any incomplete student records or those with extensive missing values. The data was then split 70/30 into training and test sets to train different classification models to estimate which applicants ultimately enroll at the university (binary response variable). R and RStudio were the main tools to build these models with the help of such packages as rpart, e1071, randomForest, glm, gbm and others.

1. First Source Analysis

The merged first source data made the disproportionate reliance of the university on cold lead sources (LEAD SOURCES) quite evident. As can be seen from the treemap below, more than 35% or 74,000 of all leads during the 2021-4 period came from such sources, with College Board, SAT and AP being the three leading sources in this category. The second largest source category with around 22% or 43,000+ leads was OTHER SOURCES, also referred to as third-party ‘hand-raiser’ channels where the student needs to express their interest. Unfortunately, this category is missing more granular information to determine the exact source for more than half of the students marked as “Other.” STEALTH APPS or applications that cannot be traced back to any lead source accounted for another 39,000+ apps or 19%. EVENTS added on another 32,000+ leads, while CASES was the smallest category with around 12,000 leads.

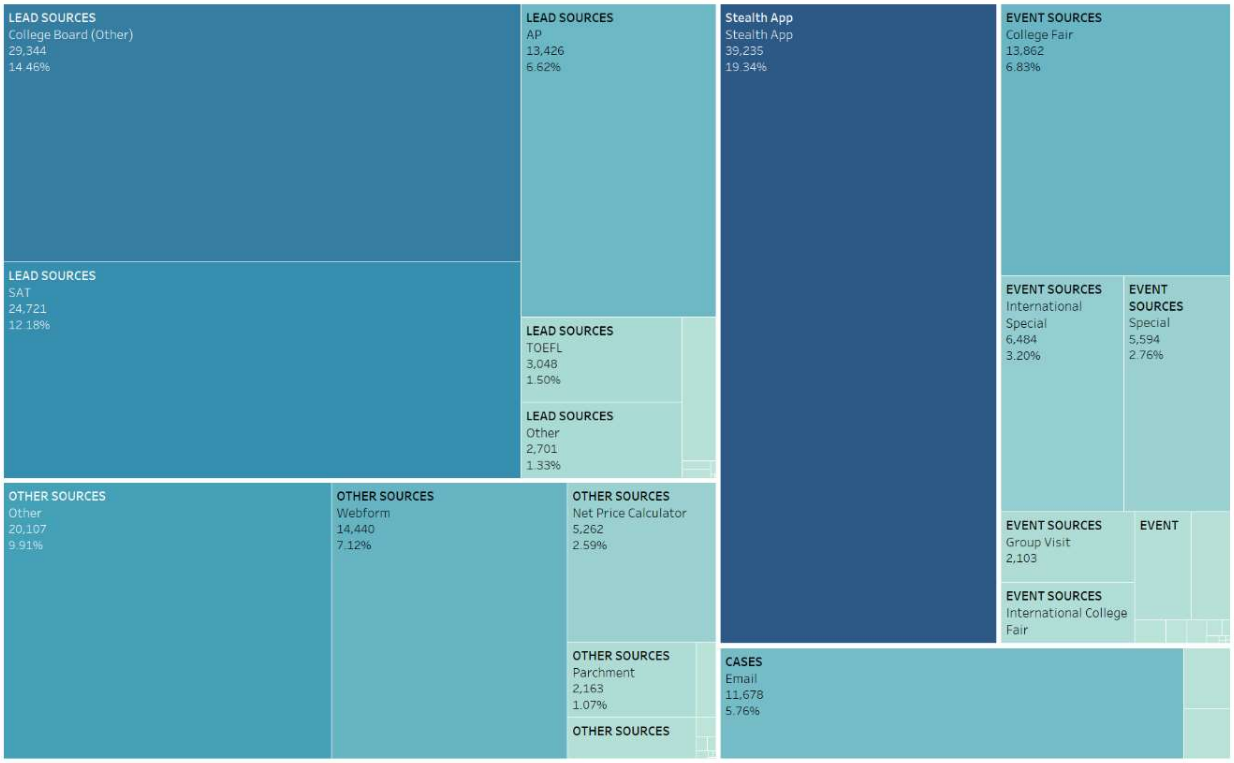


Figure 1: Lead Sources by Category, 2021-24

To explore the shifting trends in our lead sources from year to year, the below bar graph was generated. It gives a visual confirmation for the reality that our reliance on cold lead sources (LEAD SOURCES) has become unsustainable in part due to the challenges posed by data protection laws mentioned earlier. This has caused a dramatic decrease in the total number of leads available to us during the 2023 and 2024 cycles. While OTHER SOURCES and STEALTH APPS have grown over the same period, they have not yet made up for this loss of cold lead sources. In fact, the steady increase in STEALTH APPS during the last two years may have captured a portion of those students who used to enter our funnel from cold lead purchases. It is also important to note here that the 2024 source data only goes up to 6/7/2024, so it may not be a complete picture of the final numbers for this year. Regardless, it is clear that the university is faced with a much smaller pool of international students leads, which necessitates identifying ways of improving our conversion rates and tools for more robust enrollment forecasting, if we are to sustain our current size and meet goals.

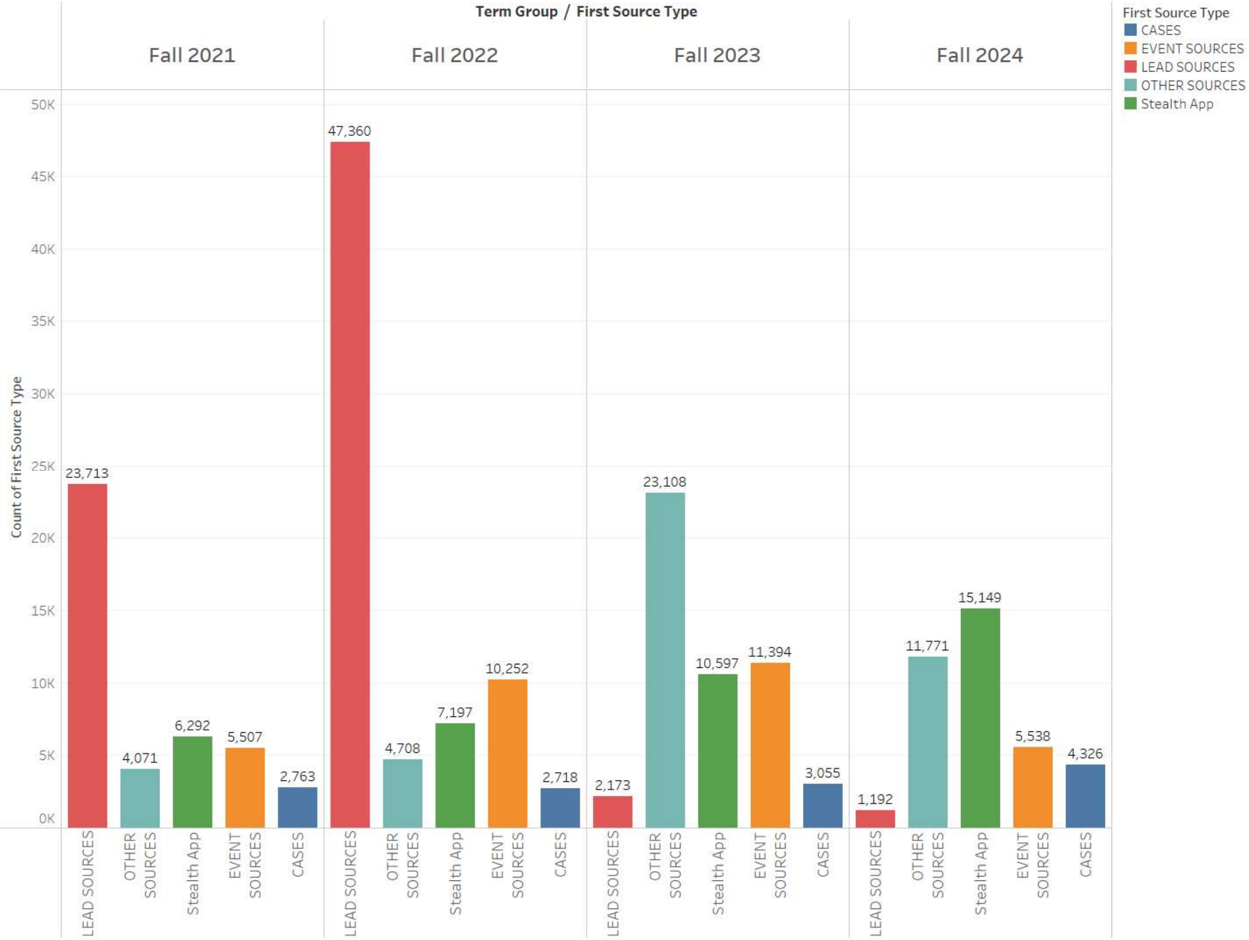


Figure 2: First Source Type Trends, 2021-24

As a final step in first source analysis, I examined our lead trends from our top markets. One of the challenges with our Salesforce data is that it is not consistent in terms of available data points for each lead. This makes it difficult to track all leads down to a specific geographic market as can be seen from the large number of unknown leads. What is most concerning from Figure 3 below is the visible decline in lead numbers from our primary markets, especially China and India which alone account for 50% of our funnel. Other markets like Saudi Arabia, UAE, Canada, Mexico, also took a major hit in 2022 and haven't fully recovered. While some of this can be attributed to Covid and economic factors, the lack of available cold lead sources undoubtedly also played a role.

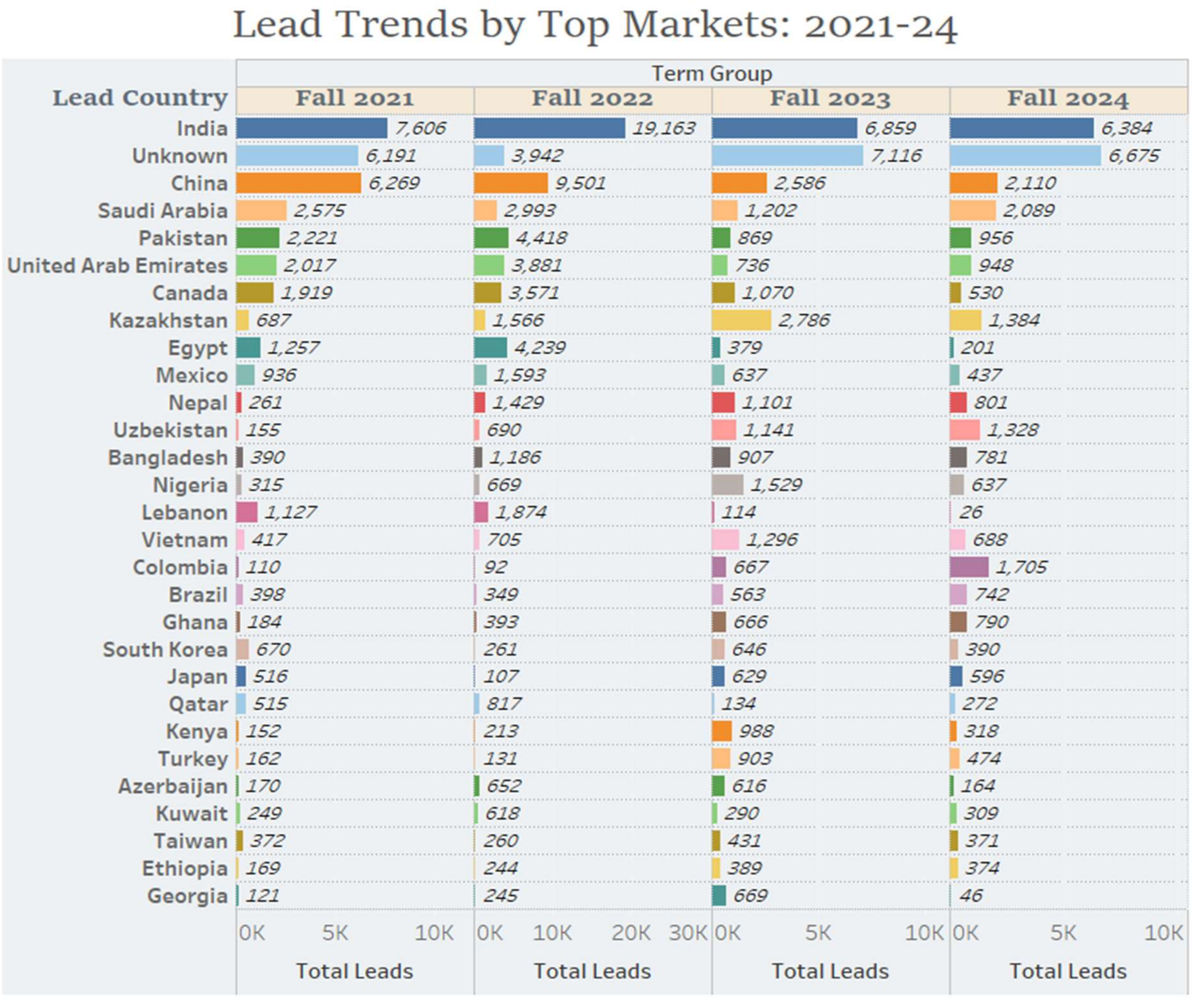


Figure 3

2. Application Data and Enrollment Prediction

The next phase consisted of merging application data with first source data. Some initial EDA was conducted to explore the university's enrollment from several angles. As the bar graph below illustrates, the enrollment rate generally fluctuates between 5% and 10% for international applicants overall. The only anomaly on the graph are international students who are already in the US and applying to the university domestically. The likely reason is that many of these students are already living in the same state and have built affinity with the university, resulting in a much higher conversion rate from applicant to enrolled student.

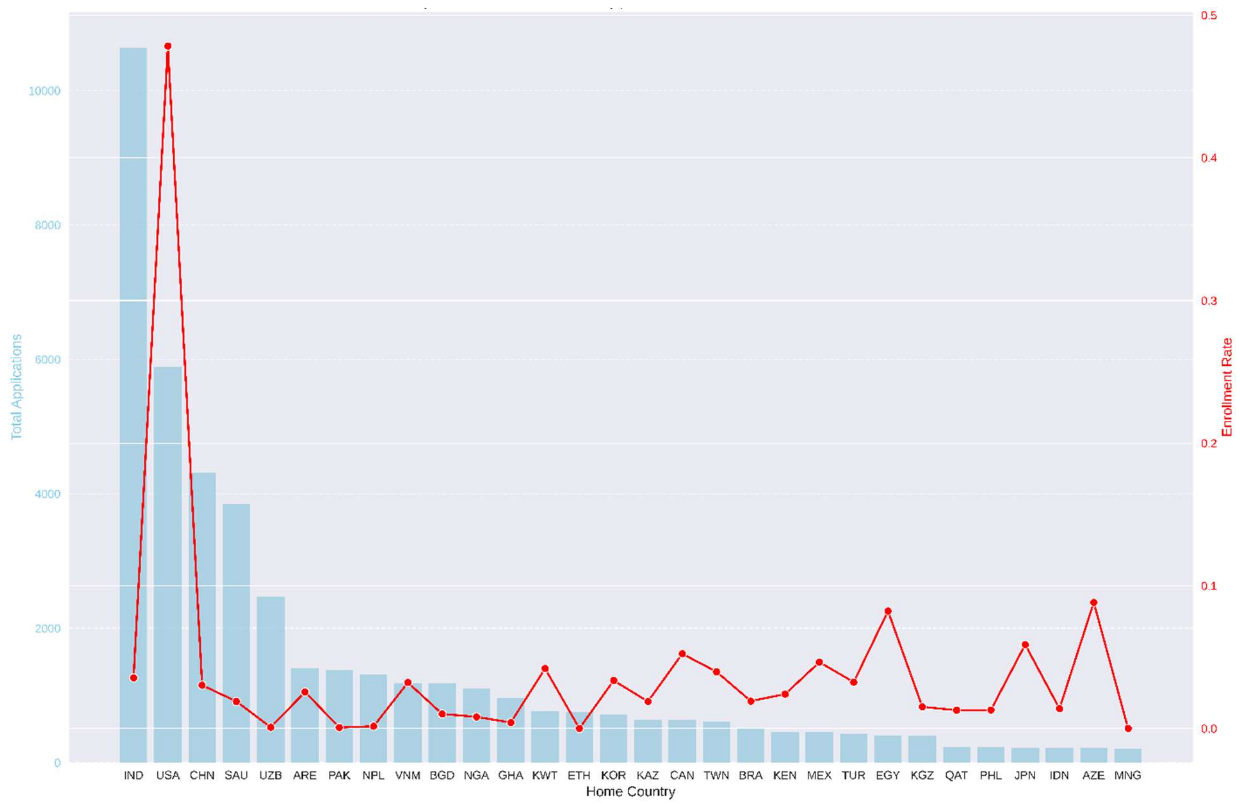


Figure 4: Application Numbers and Enrollment Rate by Country

Exploring the first lead sources including their subcategories showed that our enrollment rate is highest for EVENT SOURCES (15.8%). These are students who have attended university organized events and have established a more personal connection with us. In particular, International Special Events have an impressive 33.3% conversion rate, so this is an area we should continue investing resources in. Other subcategories performed worse or didn't contain enough data points to draw meaningful insights.

The second best category was Stealth Apps with an 11.5% enrollment rate, which represents students who found us on their own without any trackable exposure to our marketing. This category may include students who were in our database from previous years, so additional data is needed to evaluate that possibility. CASES refers to Salesforce logged inquiries from students, which generated a 5.9% conversion rate. This looks rather low considering that these are students who are already aware of our brand and proactively seeking information – we should be able to make improvements in this category.

The worst enrollment performance overall is from our LEAD SOURCES (cold leads) and OTHER SOURCES (hand-raisers) from various third-party channels. This reaffirms and exacerbates concerns regarding dwindling sources of cold and warm leads from third parties: not only are there fewer leads to purchase, but these leads are resulting in very few conversions for us in international markets. We need to be very strategic about selecting the highest performing vendors and actively looking for new and emerging platforms that may offer us better quality leads targeted to our institution.

Table 1: Enrollment Rates by Lead Source Type

First Sources	Enrollments	Total Apps	Enr Rate
CASES	**66**	**1116**	**5.9%**
Email	66	1113	5.9%
Incoming Call	0	3	0.0%
EVENT SOURCES	**74**	**468**	**15.8%**
College Fair	2	129	1.6%
Experience University	3	18	16.7%
Group Visit	1	4	25.0%
International College Fair	0	11	0.0%
International Group Visit	0	18	0.0%
International Special	55	165	33.3%
More to Explore	0	2	0.0%
Special	13	121	10.7%
LEAD SOURCES	**31**	**1306**	**2.4%**
SAT	3	369	0.8%
AP	3	129	2.3%
College Board (Other)	23	752	3.1%
EDX GFA	0	2	0.0%
PSAT	0	19	0.0%
PTK	0	3	0.0%
TOEFL	2	32	6.3%
OTHER SOURCES	**30**	**631**	**4.8%**
Cappex	0	6	0.0%
College Board (Other)	2	7	28.6%
Common App	0	3	0.0%
Net Price Calculator	3	66	4.5%
Other	4	54	7.4%
Parchment	0	5	0.0%
Raise.Me	1	1	100.0%
Unibuddy	1	16	6.3%
Webform	19	473	4.0%
Stealth App	**3707**	**32205**	**11.5%**
Stealth App	3707	32205	11.5%
Grand Total	**3908**	**35726**	**10.9%**

Predictive Models

Data Preparation

When merging the app data with first source files, nearly 6,968 records had no first source that could be identified – these were further marked as 'Stealth App.'

```
first_source_type
Stealth App      44832
LEAD SOURCES      1609
CASES             1332
OTHER SOURCES     1023
EVENT SOURCES      571
```

After aggregating the 2021-4 application data and merging the first source information to it, I was left with a list of 49,368 rows and 84 columns. However, this dataset needed further transformation to be suited for training predictive models. Several important changes using the techniques below were made to ensure that the remaining data was relevant, meaningful, and usable:

1. **Simplification:** after some deliberation, only students with completed applications were included, i.e. all students with *application_status* equal to 'Application Received – Incomplete' were deleted. Since a university cannot begin the review process of applications that are missing required documents, these students are stuck and cannot progress down the funnel to become denied, admitted or enrolled. For this reason and due to lack of other data points for this group of students, it made less sense to include them in the training or testing data for classification purposes. By definition, all students from past cycles with incomplete applications on the list have not enrolled. However, as a future extension of this project, I would like to attempt including them to produce a more realistic representation of the very top of our funnel.
2. **Feature engineering:** all datetime columns (*application_date, app_complete_date, admit_date*) were transmuted into month values to be used as categorical predictors. Additionally, they were used to create new columns to measure the time difference or distance in days between the respective date and the start of classes for that cycle (late August of each year). This is a fairly common feature engineering technique for datetime variables to extract potentially useful information because datetime values cannot be directly plugged into predictive models.[5] Both domain knowledge and intuition aided this decision: the likelihood of a student enrolling may well be related to how far in advance they apply, how quickly they complete the application, and how soon we admit them.

[5] https://scikit-learn.org/stable/auto_examples/applications/plot_cyclical_feature_engineering.html.

3. **Aggregation:** several categorical columns with infrequently occurring values were cleaned or aggregated to reduce their cardinality. For instance, *student_type* was filtered to include only FTF (Full Time Freshman) and TRN (Transfer), gender was filtered down to only M/F. The *major* field was whittled down from over a dozen categories to ten using a simple aggregating function with a threshold value to group all low frequency instances under "Other." Similarly, *age* was transformed into a binary value to signify whether the student was over 18 at the time of application.
4. **Frequency encoding:** for columns with even higher cardinality (>100 unique values), such as *home_country* and *home_city,* I chose to use Frequency Encoding. In Frequency Encoding, categories are replaced by their frequencies or counts in the dataset.[6] The frequency of a category is calculated as the number of times that category appears in the dataset. This count can be normalized by dividing by the total number of data points to represent it as a percentage or probability, which is the approach I chose. A big advantage of this technique over One-Hot Encoding is that it doesn't increase the dimensionality of the data.
5. **Imputation:** Missing data was imputed using the column means for about 250 missing GPA values and 12 *appcomp_date_diff* values; a couple of other small errors were fixed.
6. **Removed variables:** multiple sparse columns were dropped due to their very low data density, including *honors_app_date, honors_denied, honors_enrolled, ci_score_intl, enrollment_deposit_status* and others. Columns with zero or little variation in their values were also dropped, such as *market, residency, hs_name, hs_code*, etc. Moreover, all applicant ID columns (*app_id, inquiry_id, inquiry_opp_id*) were scrapped since these were anonymized values that had been previously deduped during data cleaning. These provide no predictive value since they are unique for each record.
7. **Domain knowledge:** because the last year of enrollment data for Fall 2024 was provided up to 6/7/2024, this point in the admission cycle was considered when concatenating the full dataset. Point-in-cycle refers to the current position in a fourteen-month recruitment cycle. The 2021-3 datasets were filtered to exclude students who started an application at or after the equivalent point-in-cycle, i.e. June 7th of each year.

One remaining issue with the dataset that needs to be addressed is the somewhat "rare event" nature of our admitted students who enroll. The final dataset after incorporating all the changes above contained 35,423 applicants and 3,830 enrolled students or a ratio of about 10:1. As Chiang points out, "Traditional machine learning algorithms and statistical models could be challenged to handle imbalanced data. The prediction results could be biased toward

[6] https://towardsdatascience.com/dealing-with-features-that-have-high-cardinality-1c9212d7ff1b

the majority class and lead to poor performance of the minority class. However, the minority class is often more significantly interesting in many applications."

There are multiple suggested ways of dealing with imbalanced datasets either at the data level through resampling or at the algorithm level by adjusting the misclassification costs (penalized model) or the decision threshold (probabilities). However, these are generally recommended for more extreme cases of imbalance where the occurrence ratio is at 2% or lower (Fawcett, 2016). Selecting the appropriate performance metric is also crucial. Other sources such as Allison also mention that Logistic Regression and tree-based models can still work well with unbalanced data. Thus, within the limited scope of this project I decided to include several predictive models from the above classes and used the dataset as is without resampling or other compensatory methods.

Measuring Model Performance

While model accuracy is generally the most popular metric used to evaluate classification models, it is not always reliable. This dataset is unbalanced with over-representation of one class (not enrolled) and under-representation of the other (enrolled). As a result, any classification model built will more often predict the majority class (not enrolled), thereby inflating the accuracy. In the context of enrollment management, Precision and Sensitivity (aka Recall or True Positive Rate) are two metrics that are more relevant for the purposes of evaluating the predictive potential of a model in terms of which students may ultimately enroll.

Precision is a metric that tells us about the quality of positive predictions. Out of all students predicted to enroll, how many of them actually enrolled? Sensitivity/recalls tells us about how well the model identifies true positives. Out of all the students who ended up enrolling, how many were correctly identified? Below are their formulas:

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{Recall} = \frac{TP}{TP + FN}$$

The F1 score is a combined metric that uses the weighted average or the harmonic mean of precision and recall, and maximizing the F1 score implies simultaneously maximizing both precision and recall. In addition to confusion matrices, I will primarily be using this metric to evaluate and compare the performance of my models.

EDA

Prior to training the models, I wanted to examine the variables with the help of some basic plots. The histograms of all the numeric variables reveal that none of them are normally distributed. The *app_date_diff* and *app_comp_datediff* (measuring distance in days to start of classes) do resemble multimodal distributions and this is likely due to the fairly cyclical nature of application volumes over a year period:

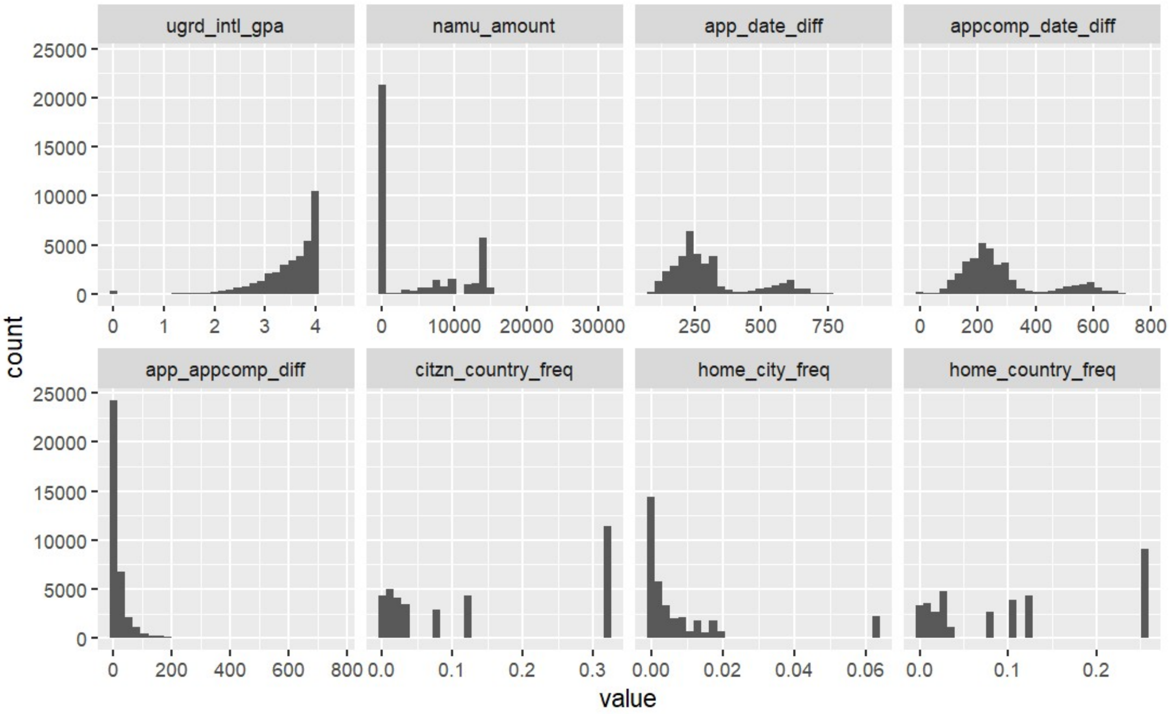


Figure 5: Histograms of Numerical Predictors

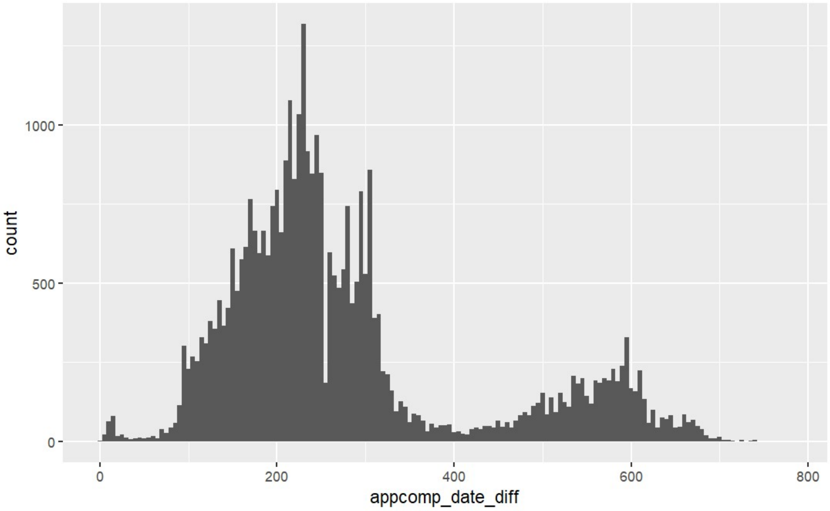


Figure 6 Expanded View of Histogram

I also created box plots of some of the more interpretable numeric variables, such as GPA and NamU scholarship amount against the response:

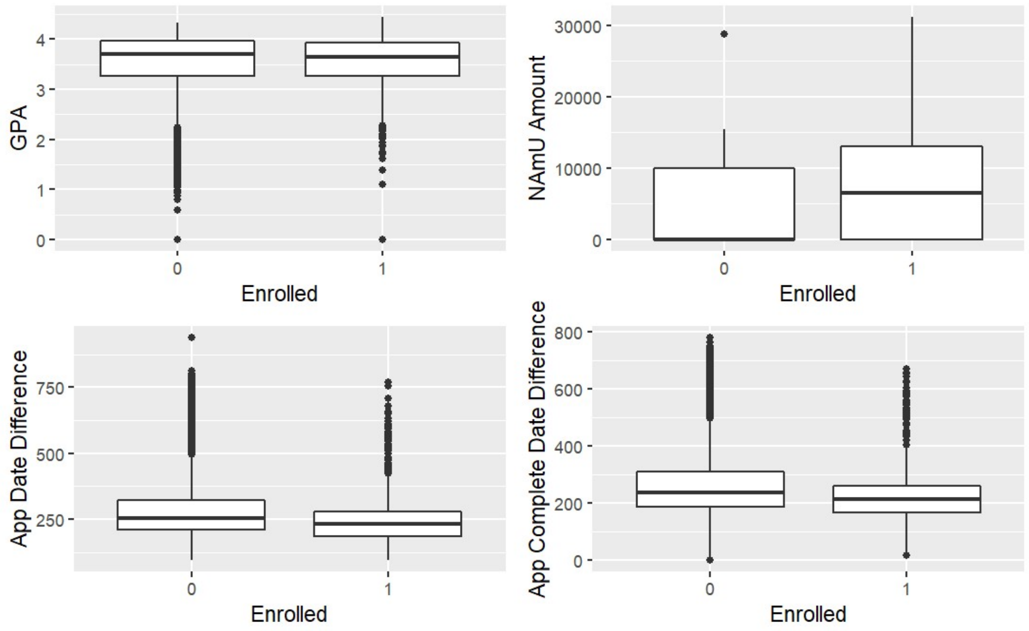


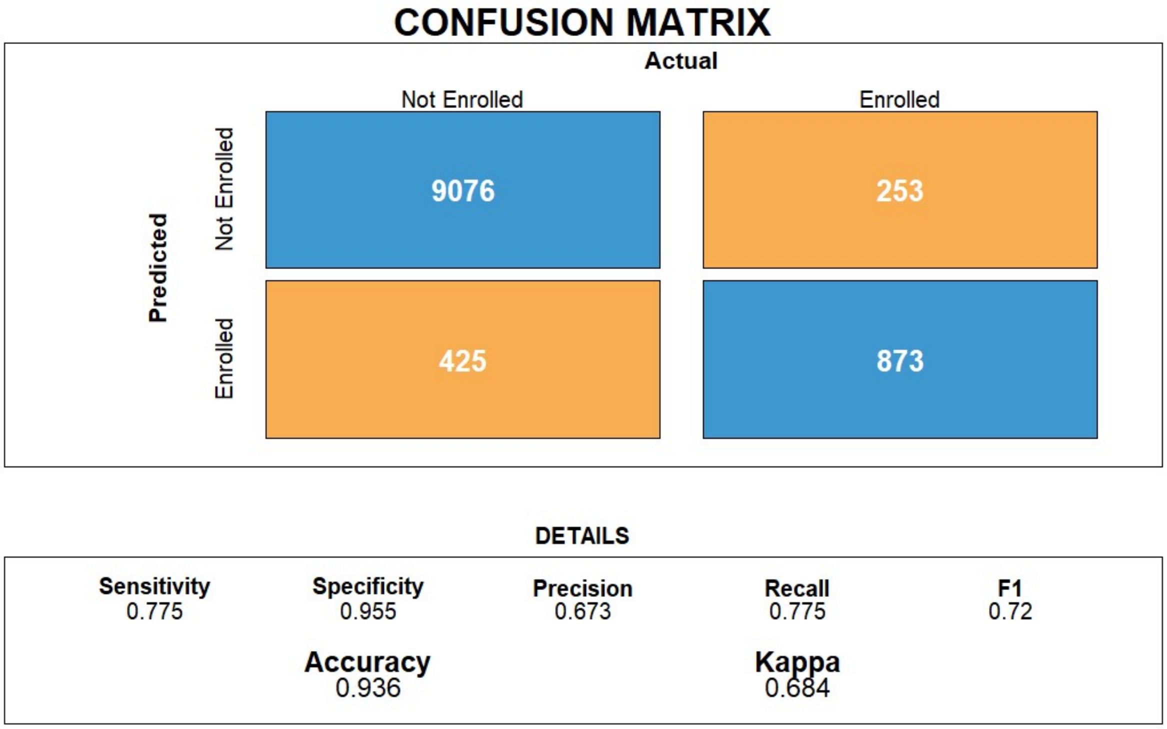
Figure 7: Box Plots

As we can see from the first plot, there doesn't appear to be a significant difference in terms of the median or the range of GPA values between students who enroll and those who don't. In part, this follows from the fact that the dataset only includes admitted students – that is, students who have met our admission requirements. Thus, GPA may not have a very noticeable effect on the student's likelihood of enrolling. On the other hand, NAmU scholarship amounts for enrolled students were clearly higher on average, which would appear to be at least a partial testament to the effectiveness of such a strategy for converting applicants, yet this is only a common sense intuition and may also be due to collinearity with other factors. As the bottom two plots show, students who enrolled also tended to apply and complete their applications a little later on average. This is an interesting signal that could warrant further investigation: it may be connected with the fact that students who apply early usually put in more preparation into their college search process and have longer college lists, whereas those who start later have fewer choices on their radar.

Model Evaluation and Comparison

I. Naïve Bayes

Using R and RStudio, the dataset was randomly sampled and split 70/30 into training and test sets. In addition to logistic regression and tree-based models, I wanted to include a Naïve Bayes model as a baseline for comparison. The Naive Bayes algorithm is simple to construct and does not require complicated parameter estimation. It gives highly interpretable results that are remarkably good in practice and remains a popular technique due to this efficiency. It is especially appropriate when the dimension of the feature space is high and assumes that the features are independent, this being the “naïve” assumption (Hastie & Tibshirani, 210-1).



CONFUSION MATRIX

Predicted \ Actual	Not Enrolled	Enrolled
Not Enrolled	9076	253
Enrolled	425	873

DETAILS

Sensitivity	Specificity	Precision	Recall	F1
0.775	0.955	0.673	0.775	0.72

Accuracy	Kappa
0.936	0.684

Despite its limitations, this model performed surprisingly well with a Recall score of 0.78 or 78%, meaning it did a fairly good job at correctly identifying students who ended up enrolling out of all enrolled students in the test set. Precision here wasn’t as ideal at 67%, with the model only correctly identifying 873 enrolled students out of the total of 425+873 that it predicted would enroll. The key for enrollment management and forecasting would be to strike a balance between these two metrics, so as not to exclude too many students that will likely enroll from our projections.

II. Single Decision Tree

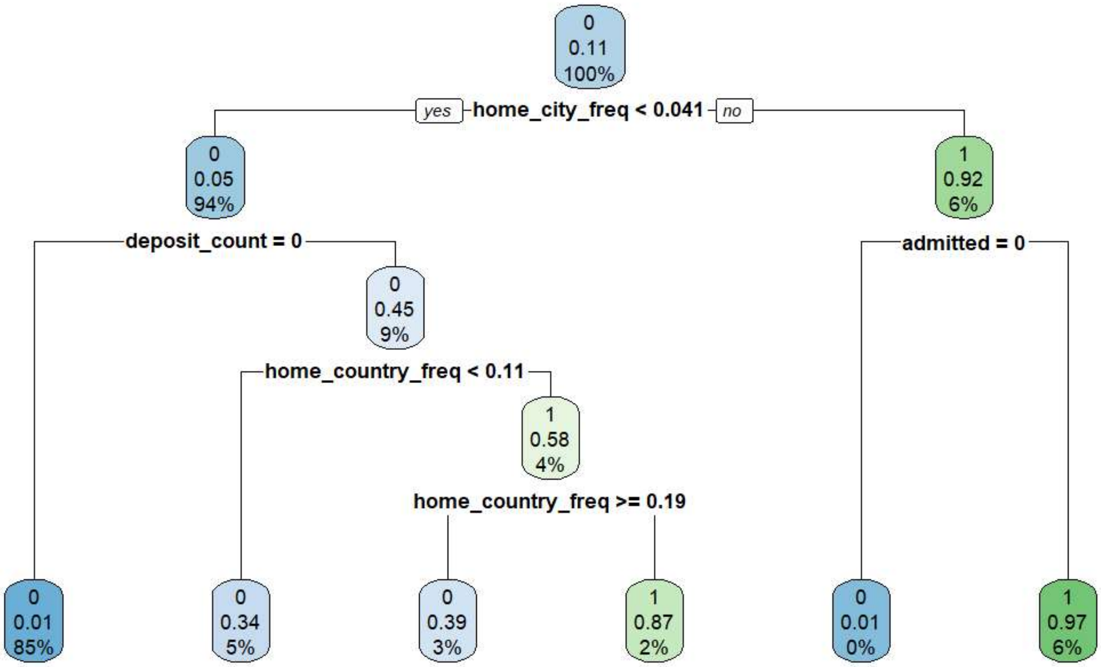
The decision tree is an algorithm that looks for optimal ways to split the data in order to provide a robust classification and regression Some key advantages of decision trees are: they often yield interpretable results that can be visually represented, making them accessible even to nonexperts, and they are capable of effectively handling both numerical and categorical data. It is helpful to briefly review the workings of this algorithm as it also applies to other tree-based models (Brunton & Kutz, 186):

(i) Scan through each component (feature) x_k (k = 1, 2, · · · , n) of the vector x_j to identify the value of x_j that gives the best labeling prediction for y_j.

(ii) Compare the prediction accuracy for each split on the feature x_j. The feature giving the best segmentation of the data is selected as the split for the tree.

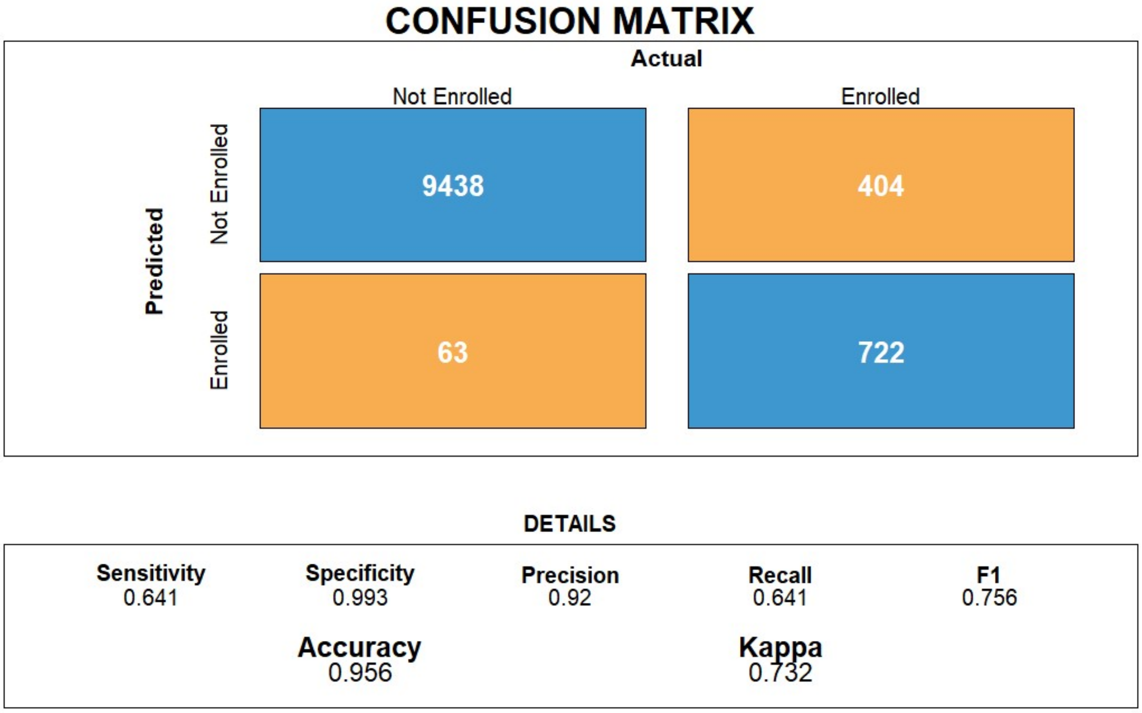
(iii) With the two new branches of the tree created, this process is repeated on each branch. The algorithm terminates once each individual data point is a unique cluster, known as a leaf, on a new branch of the tree.

Using the rpart library, we can not only easily train the model but also visualize the tree structure that shows some of these splits or nodes:



The initial tree was pruned using the optimal CP value that rpart calculates automatically through 10-fold cross validation. This is generally a good idea because full grown trees are over-fitted and don't perform well against data that is not in the training set.

We can see that the first, third, and fourth level splits are all using *home_city_freq*, so it is an influential variable within the model. The next notable split happens at admitted=0, where we see that if this is NOT the case (student IS admitted), then they are an enroll (1) with a probability of 97% and this represents about 6% of the full dataset. Similarly, on the left side the split happens at *deposit_count*, which predicts 0 (not enrolled) to the left with a terminal node representing 85% of the dataset, as well as 0 to the right with higher probabilities and accounting for a smaller portion of the data. The tree continues growing with two additional splits from there. The deposit status appearing as an important factor here is not so surprising, as a monetary commitment in the form of a deposit is a commonly used tool to drive conversions in the realm of admissions and other industries. What is more useful about this model is its ability to further distinguish among the students to the right of that split – those who have paid the deposit, but may still choose not to enroll.



CONFUSION MATRIX

Predicted	Actual: Not Enrolled	Actual: Enrolled
Not Enrolled	9438	404
Enrolled	63	722

DETAILS

Sensitivity	Specificity	Precision	Recall	F1
0.641	0.993	0.92	0.641	0.756

Accuracy	Kappa
0.956	0.732

Opposite to Naïve Bayes, this Decision Tree model performed better on Precision than Recall, meaning that the percentage of actual enrolls from its pool of predicted enrolls was higher: 722/(63+722) = 0.92; it did not fare as well for correctly classifying as many enrolled students out of the actual enrolled pool: 722/(404+722) = 0.641.

III. Random Forest

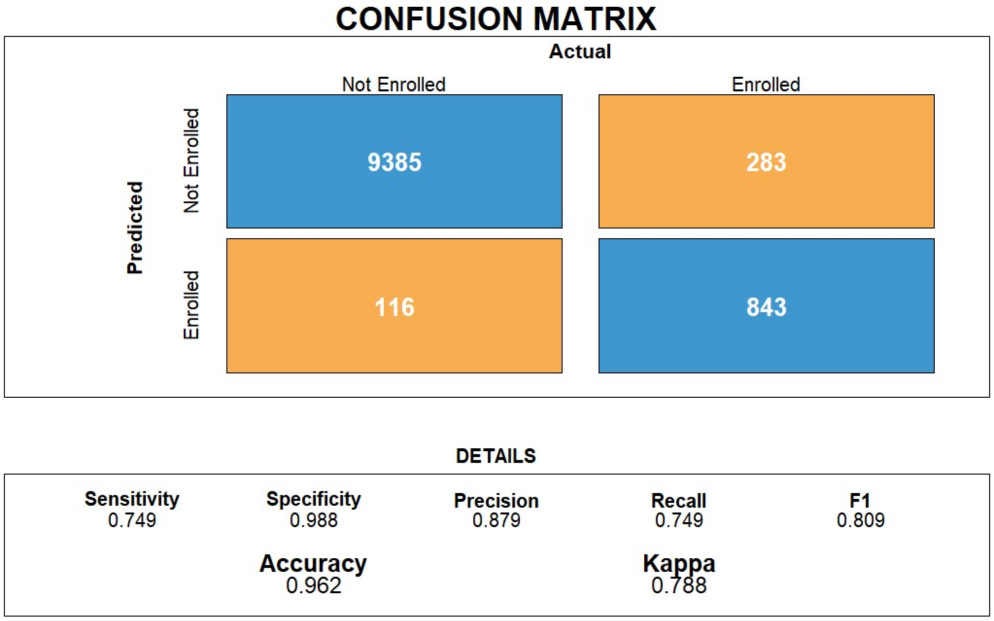
Random Forests are an ensemble learning technique that is “a substantial modification of bagging that builds a large collection of de-correlated trees, and then averages them” (Hastie, 587). This is an important innovation because individual decision trees created by splitting are generally not a good fit to different samples of the data. Thus we can generate two significantly different classification trees with two subsamples of our dataset. This presents significant challenges for cross-validation. Instead, we construct a multitude of decision trees during the training process. The random decision forests correct for any particular decision trees’ habit of overfitting to its training set, thereby giving us a much more robust framework for classification than a single tree (Brunton, 189-90). Another great built-in feature of this algorithm is its use of out-of-bag or OOB samples, providing an OOB error estimate which is nearly identical to that obtained by N-fold cross validation.

To better evaluate the predictive potential and viability of this model for students at different stages of our funnel, I decided to build using two scenarios: one with the *deposit_flag* variable included and one without it to see how the performance would compare. Depositing is usually a significant event in the recruitment funnel as it signifies the student has committed to the university. Although this step is by no means a guarantee of enrollment, it is commonly viewed by admission offices as a factor highly correlated with a positive outcome. Thus, it would be helpful to see whether models without such information can still provide comparable predictive utility.

RF with deposit flag

```
randomForest(formula = enrolled ~ ., data = apps.train, ntree = 500)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 5
        OOB estimate of  error rate: 3.46%
Confusion matrix:
      0    1 class.error
0 21844  248  0.01122578
1   611 2093  0.22596154
```

As can be seen from the function output above, the OOB error rate (3.46%) is in fact a very close approximation of the test error rate below: 1 – Accuracy = 0.38 or 3.8%). Again, this is due to its similarity with cross validation, which RF performs while training the model.



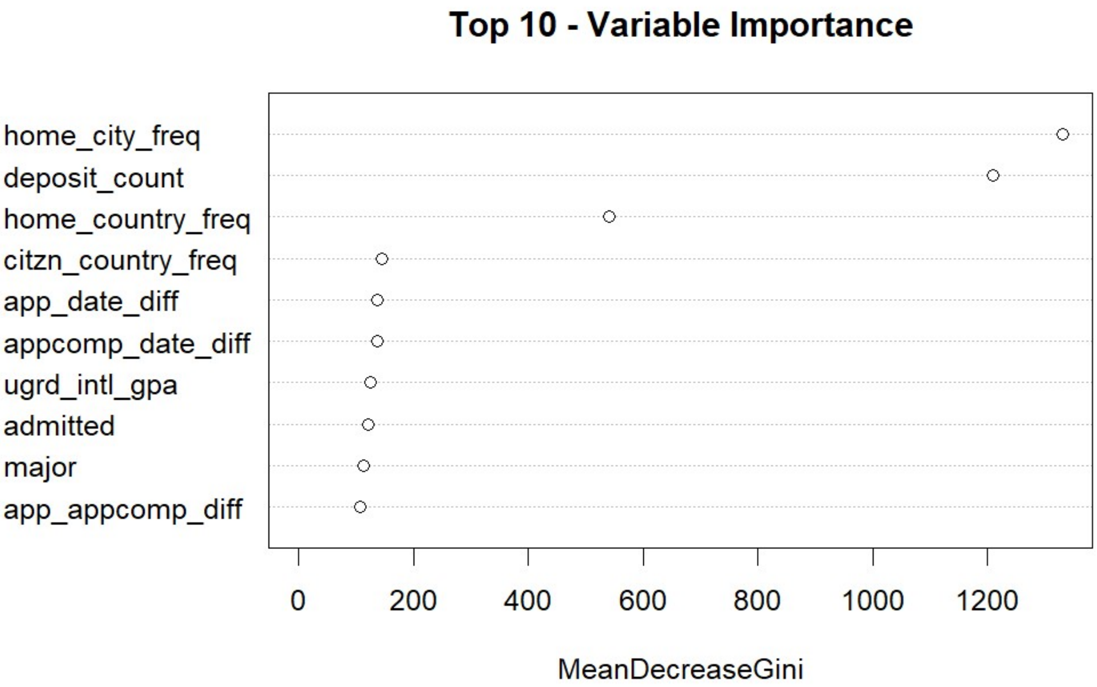
CONFUSION MATRIX

Predicted \ Actual	Not Enrolled	Enrolled
Not Enrolled	9385	283
Enrolled	116	843

DETAILS

Sensitivity	Specificity	Precision	Recall	F1
0.749	0.988	0.879	0.749	0.809

Accuracy	Kappa
0.962	0.788

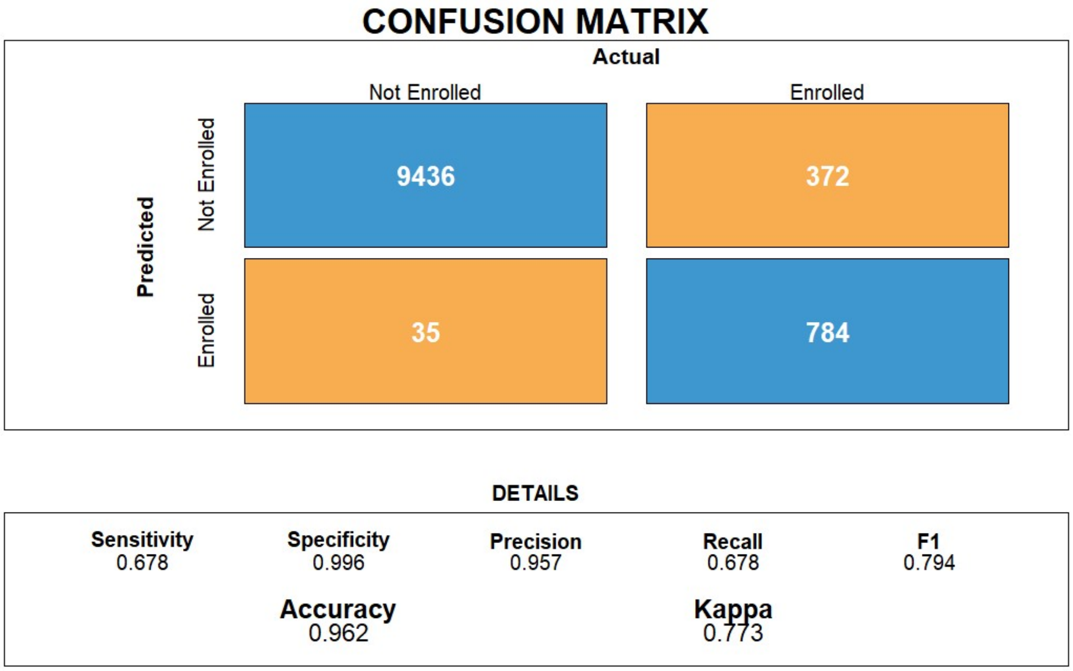


This model had a high F1 score of nearly 0.81, striking a rather good balance between precision and sensitivity. The plot above is especially illustrative for understanding how each variable contributes statistically to predicting a classification outcome. We see that the factors of *home_city_freq*, *deposit_count* and *home_country_freq* have the highest predictive value in correctly classifying enrollment. Let’s attempt the same model without the deposit status information.

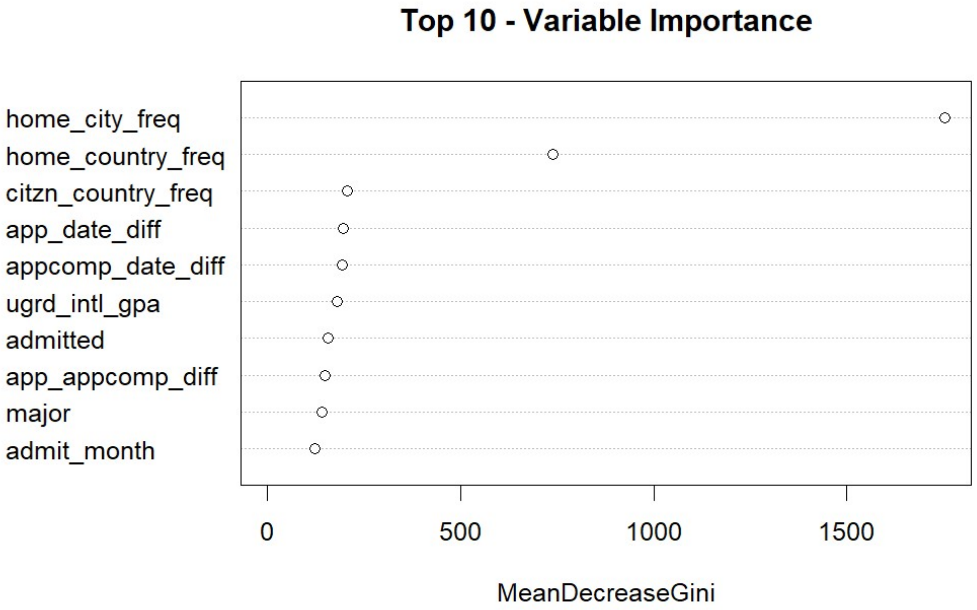
RF without deposit_flag

While not too far behind on the F1 score, this model presents a less balanced combination of recall and precision. Recall here suffered with the model identifying a smaller portion of the total enrolled pool correctly as enrolled: 784/(372+784) = 0.678; precision is astonishingly high with nearly 96% of all enrolled=1 classifications being accurate.

The variable importance plot bears a striking resemblance to the one above, with the exclusion of *deposit_count* from the list. It shows that of our champion random forest model without the deposit column, the factors of *home_city_freq* and *home_country_freq* have the highest predictive value in determining enrollment. It is a notable fact that the home city and country frequencies dominated in both models. This seems to suggest that the more frequently occurring cities and countries in our applicant pool carry important signals about the chances of enrollment. This also makes sense intuitively when examining Figure 4 on page nine and looking at the enrollment rates by country. Although a more thorough analysis is required, it is likely that we are seeing higher conversion rates, on average, from our larger markets at both country and city levels.



Sensitivity 0.678
Specificity 0.996
Precision 0.957
Recall 0.678
F1 0.794
Accuracy 0.962
Kappa 0.773

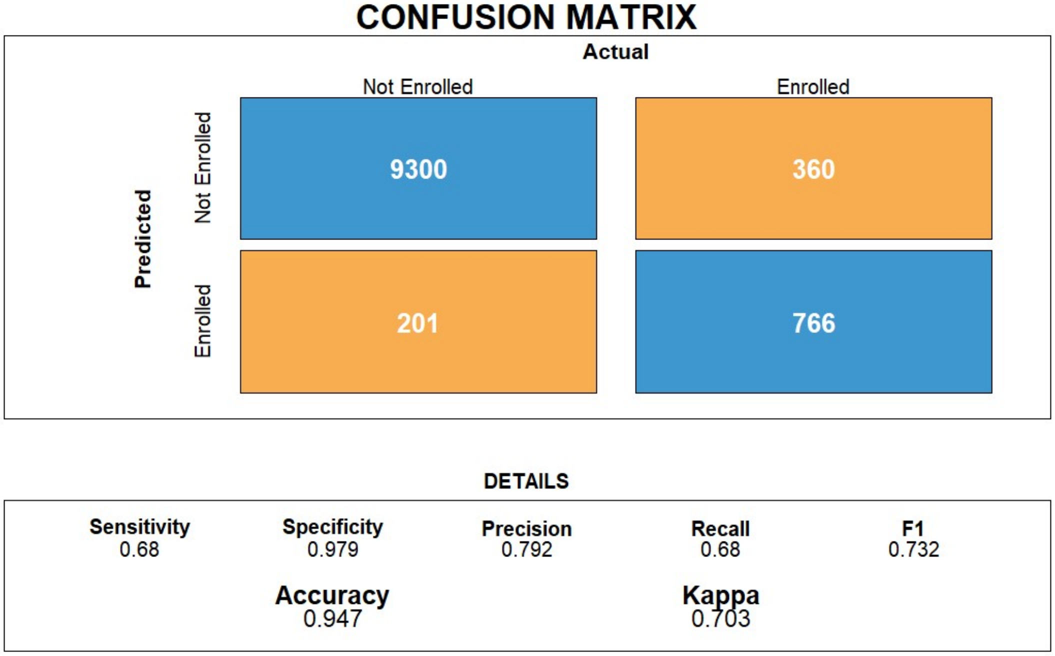


IV. *Logistic Regression*

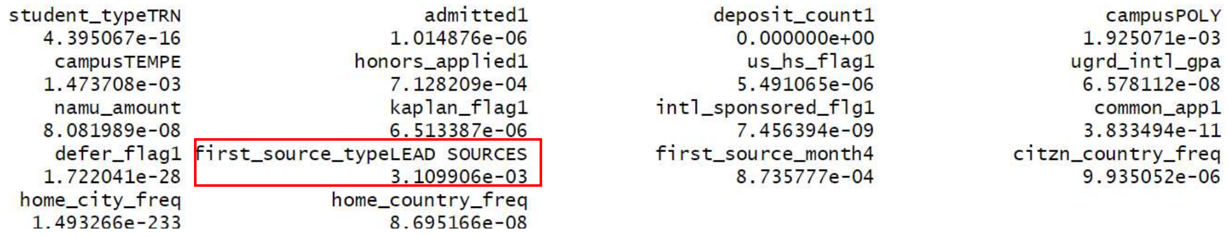
The next classification model I wanted to try was Logistic Regression, a very well-studied linear classifier technique. While this model is relatively straightforward to train, optimizing it and interpreting the coefficients is more challenging, particularly when variable collinearity is present in the dataset (Gelman, 220). Despite this, there are ways of dealing with such issues. More importantly, "logistic regression appears to have a consistent performance along the imbalance scenarios…it is safe to say that a good baseline model for imbalanced classification problems is a logistic regression." [7] The glm library and function was used in R to train two models: using all 32 available predictors and a second model that excluded *deposit_count*.

[1] https://pibieta.github.io/imbalanced_learning/notebooks/pablo-baseline-experiment.html

With deposit_count



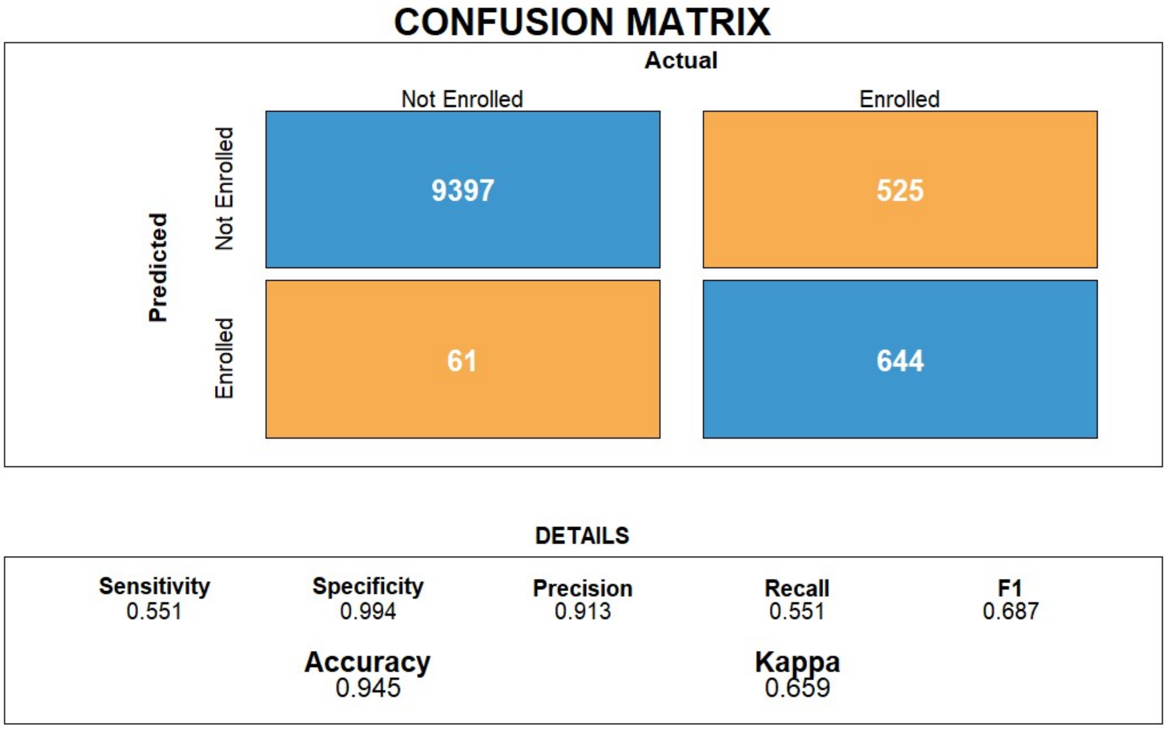
This default full model exhibits a good balance between recall and precision, but the overall F1 score is lower than most previous models, only slightly surpassing Naïve Bayes. Coefficients that were statistically significant at level 0.01 or 1% using p-values:



```
 student_typeTRN                admitted1            deposit_count1              campusPOLY
    4.395067e-16             1.014876e-06              0.000000e+00            1.925071e-03
     campusTEMPE           honors_applied1                us_hs_flag1           ugrd_intl_gpa
    1.473708e-03             7.128209e-04              5.491065e-06            6.578112e-08
     namu_amount              kaplan_flag1         intl_sponsored_flg1             common_app1
    8.081989e-08             6.513387e-06              7.456394e-09            3.833494e-11
     defer_flag1 first_source_typeLEAD SOURCES      first_source_month4       citzn_country_freq
    1.722041e-28             3.109906e-03              8.735777e-04            9.935052e-06
  home_city_freq        home_country_freq
   1.493266e-233             8.695166e-08
```

The interpretation of the p-value associated with a predictor variable is that if it is less than the chosen significance level (e.g., 0.05), you can reject the null hypothesis that the coefficient for that variable is zero. This suggests that the variable is statistically significant and has a non-zero effect on the response (Hilbe). For instance, all other variables being held constant, a positive value of the *first_source_typeLEAD_SOURCES* coefficient above would increase the odds of a student enrolling by exp(3.109906e-03) = 1.00311 or 0.311%. Most of the coefficients in this model have very small values that do not have large effects on the odds of enrollment, which makes them a little challenging to interpret individually despite them being statistically significant. The highest coefficients by relative value are *first_source_typeLEAD_SOURCES*, *campusTEMPE*, and *campusPOLY.*

Without deposit_count



CONFUSION MATRIX

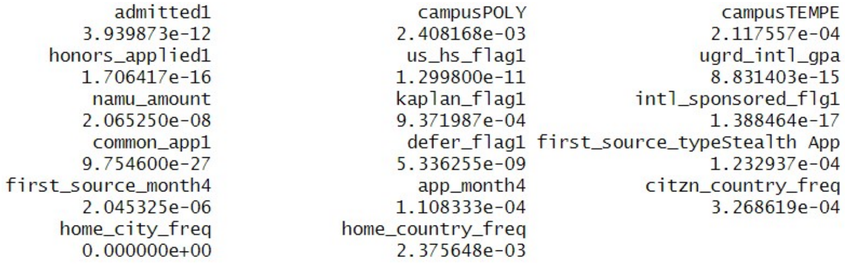
Predicted \ Actual	Not Enrolled	Enrolled
Not Enrolled	9397	525
Enrolled	61	644

DETAILS

Sensitivity	Specificity	Precision	Recall	F1
0.551	0.994	0.913	0.551	0.687

Accuracy	Kappa
0.945	0.659

Similar to the Random Forest case above, the model missing the deposit information had a comparatively lower F1 metric – while improving on precision it gave up quite a lot in recall. Coefficients statistically significant at level 0.01 (p-value) were:



```
          admitted1              campusPOLY              campusTEMPE
       3.939873e-12            2.408168e-03             2.117557e-04
     honors_applied1            us_hs_flag1             ugrd_intl_gpa
       1.706417e-16            1.299800e-11             8.831403e-15
         namu_amount            kaplan_flag1       intl_sponsored_flg1
       2.065250e-08            9.371987e-04             1.388464e-17
         common_app1             defer_flag1 first_source_typeStealth App
       9.754600e-27            5.336255e-09             1.232937e-04
 first_source_month4              app_month4        citzn_country_freq
       2.045325e-06            1.108333e-04             3.268619e-04
      home_city_freq       home_country_freq
       0.000000e+00            2.375648e-03
```

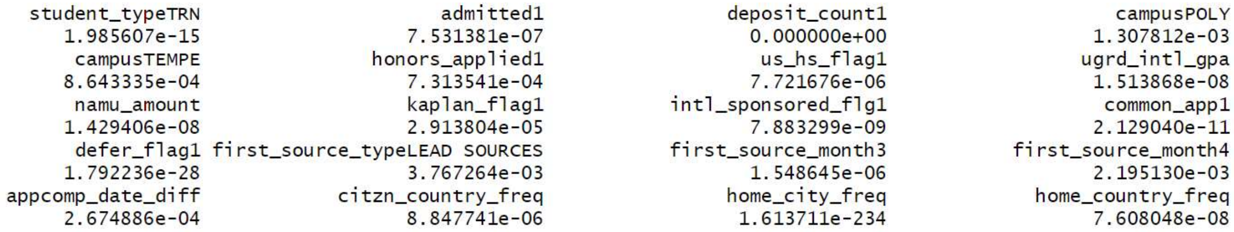
It is interesting to note some of the changes in the above list as both the coefficient values change, as well as some of the variables. For example, *first_source_Stealth App* appears on the list while *student-typeTRN* disappears from it. The highest coefficients by relative value are *home-country-freq* and *campusPOLY*.

Stepwise Logistic Regression

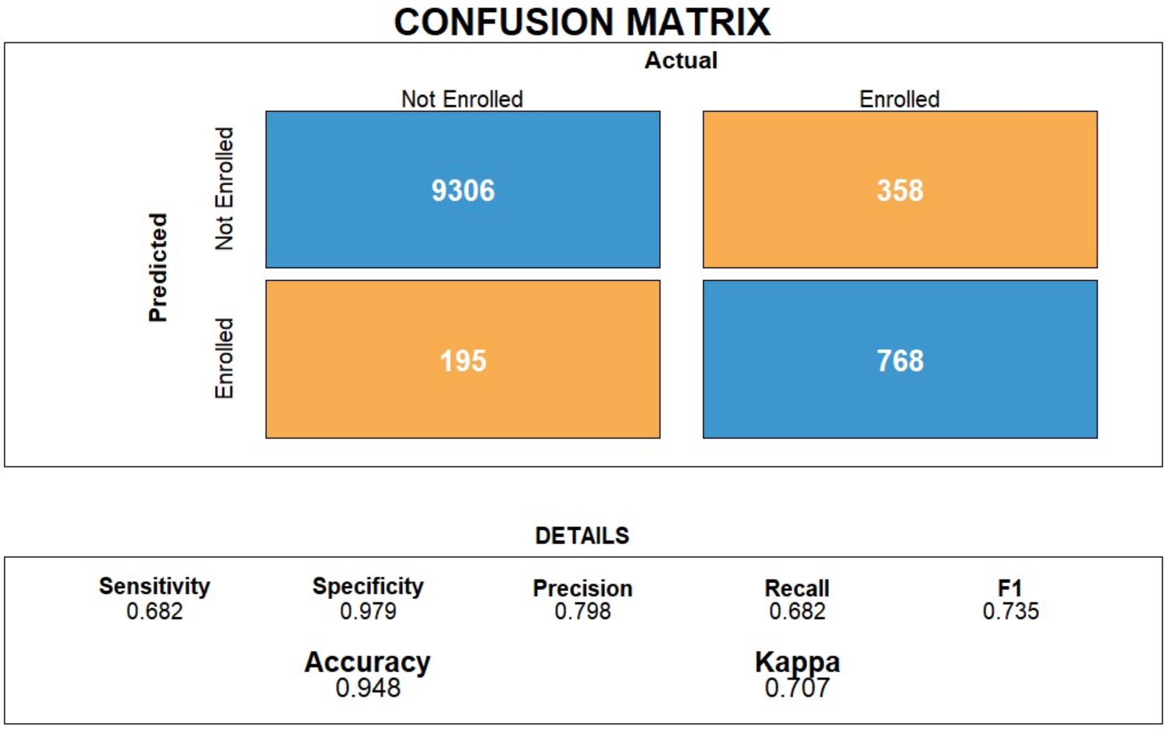
As a final step and to optimize the model, I performed stepwise logistic regression, which iteratively removes variables from the full model to balance optimal performance with using the least possible number of predictors. StepAIC function from the MASS package was used for this. The model with the lowest AIC score was as below using only 23 of the 32 predictors.

```
formula = enrolled ~ student_type + admitted + deposit_count +
    campus + degree_type + honors_applied + us_hs_flag + postsec_flag +
    ugrd_intl_gpa + namu_amount + kaplan_flag + intl_sponsored_flg +
    intl_financial_guarantee_flg + common_app + defer_flag +
    first_source_type + first_source_month + app_comp_month +
    admit_month + appcomp_date_diff + citzn_country_freq + home_city_freq +
    home_country_freq
```

Coefficients statistically significant at level 0.01 (p-value)



```
  student_typeTRN                admitted1                deposit_count1                campusPOLY
     1.985607e-15             7.531381e-07                  0.000000e+00              1.307812e-03
      campusTEMPE           honors_applied1                  us_hs_flag1              ugrd_intl_gpa
     8.643335e-04             7.313541e-04                  7.721676e-06              1.513868e-08
      namu_amount              kaplan_flag1          intl_sponsored_flg1                common_app1
     1.429406e-08             2.913804e-05                  7.883299e-09              2.129040e-11
      defer_flag1 first_source_typeLEAD SOURCES       first_source_month3        first_source_month4
     1.792236e-28             3.767264e-03                  1.548645e-06              2.195130e-03
appcomp_date_diff        citzn_country_freq                home_city_freq          home_country_freq
     2.674886e-04             8.847741e-06                 1.613711e-234              7.608048e-08
```



CONFUSION MATRIX

Predicted \ Actual	Not Enrolled	Enrolled
Not Enrolled	9306	358
Enrolled	195	768

DETAILS

Sensitivity	Specificity	Precision	Recall	F1
0.682	0.979	0.798	0.682	0.735

Accuracy	Kappa
0.948	0.707

While at first glance this model looks as if it didn't improve much on the full model (0.735 vs. 0.723 F1), it effectively maintained a similar level of predictive power while cutting out nearly 10 predictors. *first_source_typeLEAD_SOURCES* and *campusPOLY* remained two of the largest coefficients among the statistically significant predictors, with *first-source-month4* also increasing in relative value. Despite its respectable predictive performance, it remains difficult to extract actionable insights from the linear regression models above due to the large number of predictors and potential interactions among them.

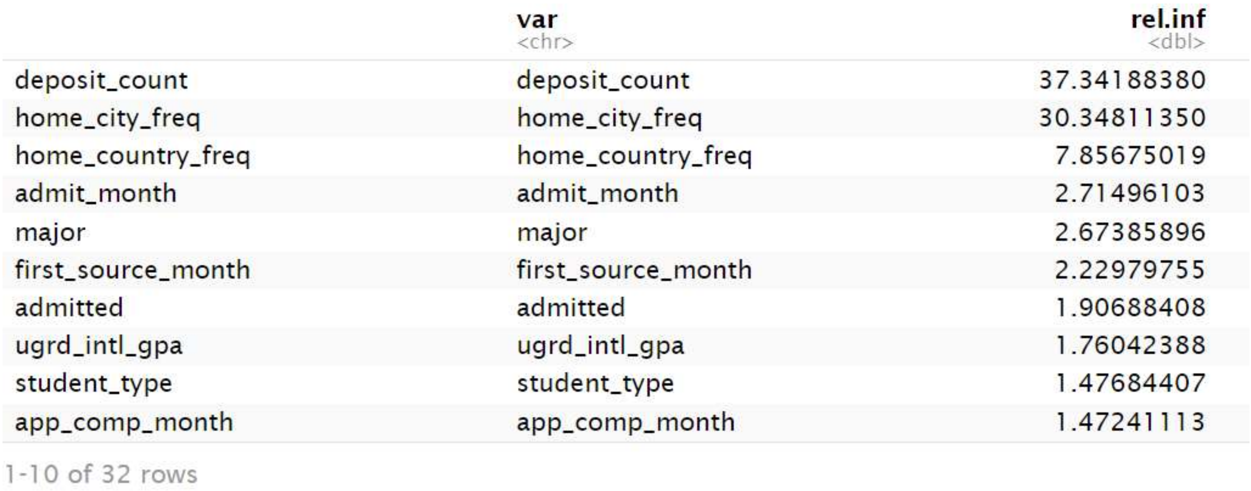
V. Gradient Boosting

Boosting is an ensemble machine learning technique that combines multiple weak models to produce a strong predictive model. Two popular boosting algorithms are gradient boosting and AdaBoost.[8] This model was trained using the gbm library in R, which is an implementation of extensions to the gradient boosting machine and AdaBoost algorithms. There are several tuning parameters that can be optimized, including interaction depth, shrinkage, number of trees and others. I have left these at default values, but further improvements could be attempted with the help of the caret package in R. This process is rather resource intensive and takes a long time to run on a local machine. The defaults used for this model were as follows:

n.trees = 5000, shrinkage = 0.05, interaction.depth = 3, cv.folds = 10

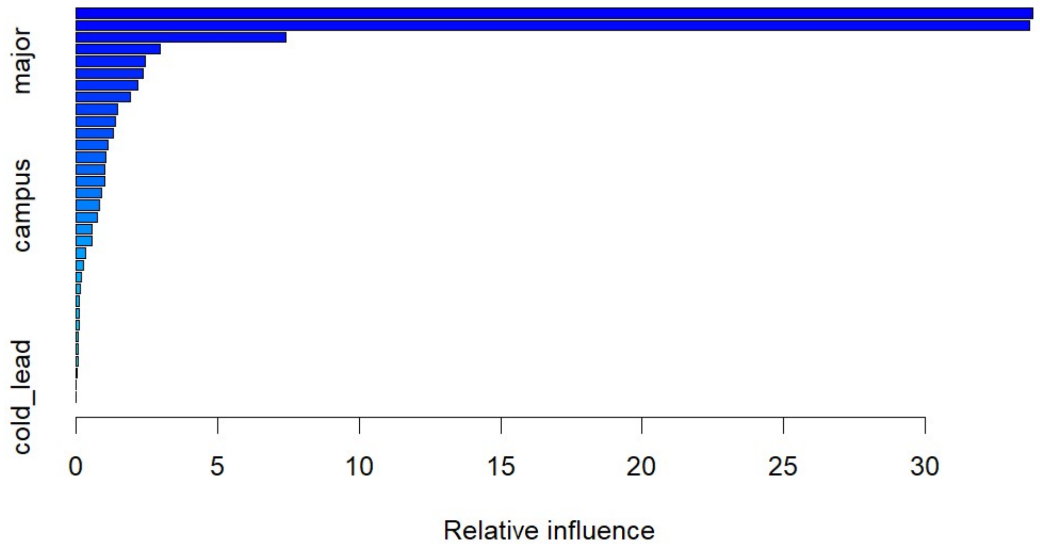
Much like RF, boosting can generate a list of variables in the model by their relative importance. Below, we see the usual suspects in the top three, although deposit status has taken the first spot. The order of the remaining predictors has also changed with *admit_month* and *major* being more influential in this model.

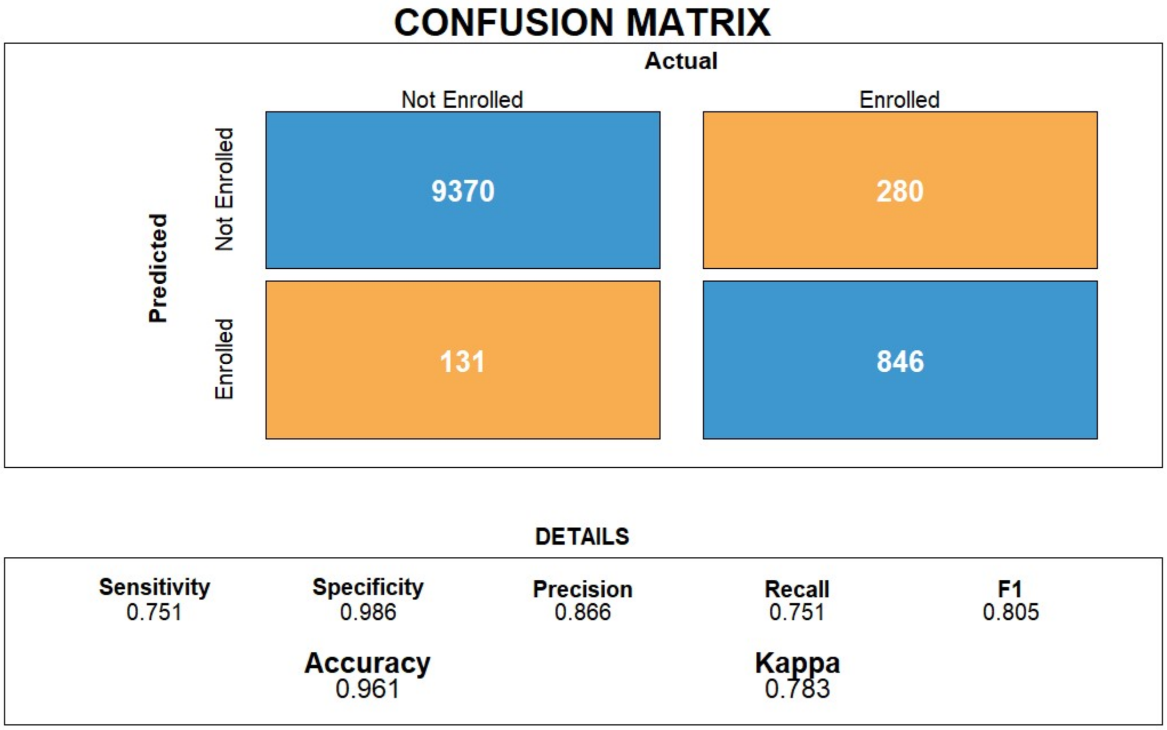
[8] https://dataheadhunters.com/academy/gradient-boosting-vs-adaboost-battle-of-the-algorithms/



	var <chr>	rel.inf <dbl>
deposit_count	deposit_count	37.34188380
home_city_freq	home_city_freq	30.34811350
home_country_freq	home_country_freq	7.85675019
admit_month	admit_month	2.71496103
major	major	2.67385896
first_source_month	first_source_month	2.22979755
admitted	admitted	1.90688408
ugrd_intl_gpa	ugrd_intl_gpa	1.76042388
student_type	student_type	1.47684407
app_comp_month	app_comp_month	1.47241113

1-10 of 32 rows





CONFUSION MATRIX

Predicted \ Actual	Not Enrolled	Enrolled
Not Enrolled	9370	280
Enrolled	131	846

DETAILS

Sensitivity	Specificity	Precision	Recall	F1
0.751	0.986	0.866	0.751	0.805

Accuracy	Kappa
0.961	0.783

This model performed rather well on the test set with an F1 score of 0.805, which is very close the full RF model.

Cross Validation

As a final step to better evaluate the performance of each model and provide a comparison, Monte Carlo Cross Validation was performed with 100 runs of each model, creating a new random 70/30 training/testing data split each time. Below are the average F1 scores and their variance for each model:

	Single Tree	**Naive Bayes**	**Logistic Reg**	**RF**	**GBM**
F-1 mean	0.7770100	0.7376882	0.7474948	0.8265000	0.8191411
F-1 var	0.00005221	0.00006445	0.00006935	0.00001952	0.00003558

Table: Average F-1 scores by model type over 100 CV iterations

The winner model on average in terms of highest F1 score is Random Forest, with Boosting (GBM) coming in as a close second. RF also had the least variance in its performance among all 5 model classes.

Conclusion and Recommendations

First Lead Sources

The analysis of first source data made clear the need for a sustained effort in finding new sources of cold leads to make up for the substantial losses incurred over the last two years. With our existing leads from EVENT SOURCES, we see great conversion rates and it would be prudent to continue capitalizing on this category to attract more students to our events in international markets to build those personal connections and a tangible connection with the university. At the same time, there is room for improvement with converting our leads from Salesforce cases. If students are seeking us out, we must make sure we are using these opportunities to enhance their perceptions of us and their interest in choosing our university. Such improvements could come from reducing our case turnaround time, but also from having more consistency in case assignment. This is still a significant pain point for our teams as often we are too busy to answer in a timely manner or students get bounced around between different staff members which disrupts the continuity of our service. Hopefully, the recent partnership with OpenAI will also integrate ChatGPT capabilities for faster and more accurate responses to student inquiries in Salesforce. Finally, we need to continually look for new third party vendors and lead generation platforms that may offer us better quality leads tailored to the university's needs and goals. For a more comprehensive analysis, lead sources need to be tracked and recorded more reliably both in terms of their geographic and vendor information.

Enrollment Prediction and Forecasting

Although the scope of this project was not extensive, it revealed the potential that tree-based ensemble methods like Boosting and Random Forest have for predicting enrollment likelihood, as well as identifying the most influential factors among the dozens of datapoints we have for each student lead. While the inclusion of deposit status in most models was generally beneficial, performance of RF even without this variable was still robust. There are several avenues that can be explored to further improve model performance that were not attempted here:

- The default probability threshold of 0.5 for prediction can be adjusted depending on specific needs or the time in cycle. For instance, we may want to bump this up to a higher value when forecasting for housing and class demand later in the cycle.
- Parameter tuning can further improve model performance, although it can be quite resource-intensive. Shrinkage, the number of trees in each forest, the minimum number of nodes, interaction depth and others can be compared using a grid search. For linear models like Logistic Regression, L1 regularization or LASSO could be attempted to better pinpoint the most influential predictors.

- Additional ways of handling high cardinality nominal variables, such as home city, major, and others, can be explored using sampling techniques or CatBoost, which is recognized for its capability of dealing with such datasets and achieving high performance.
- In addition to Precision, Recall and F1, model performance can be compared by using the AUC ROC curves and the Brier score, which also factors in the individual probabilities of each prediction.
- If identifying students who will not enroll (true negatives) becomes more important in certain applications, we can revert to using specificity and precision.

It must be emphasized that more robust information collection and aggregation would greatly enhance our predictive modeling ability. The following data points may be quite helpful for making more accurate and nuanced predictions:

- High school size, curriculum, tuition
- A consistent scaled EPR score for each student regardless of the type of test
- Precise lead source type and name
- Email campaign clickthrough rate
- Student financial resources and ability to pay

As a continuation of this project, there is still plenty of room for exploring additional models that have use cases for various stages of the funnel: cold lead to applicant, cold lead to deposit, deposit to enroll. Some of these models could take advantage of the larger lead data that includes non-converted leads or students who didn't apply or did not complete their application. These models would allow for more nuanced predictions and enrollment estimates at different points in the recruitment cycle. Other interesting extensions could involve using Poisson Regression to model our daily application numbers throughout the year, as well as using simulation modeling to optimize our application processing or other areas like housing capacity.

Citations

Allison, P. (2012). Logistic regression for rare events. Statistical Horizons. https://statisticalhorizons.com/logistic-regression-for-rare-events/

Brunton, S. L., & Kutz, J. N. (2019). Data-driven science and engineering: Machine learning, dynamical systems, and control. Cambridge University Press.

Chiang, J. Y., Lio, Y., Hsu, C. Y., Ho, C. L., & Tsai, T. R. (2023). Binary classification with imbalanced data. Entropy, 26(1), 15. https://doi.org/10.3390/e26010015

Fawcett, T. (2016). Learning from imbalanced classes. SVDS. https://www.svds.com/learning-imbalanced-classes/

Gelman, A., Hill, J., & Vehtari, A. (2020). Regression and other stories. Cambridge University Press.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2019). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.

Hilbe, J. (2015). Practical guide to logistic regression. Chapman & Hall.

Khan, A. A. (2022). Balanced split: A new train-test data splitting strategy for imbalanced datasets. arXiv. https://doi.org/10.48550/arXiv.2212.11116

DataSciencetest. (2023). Managing unbalanced classification problems. https://datascientest.com/en/management-of-unbalanced-classification-problems-ii

QS Quacquarelli Symonds Limited. (n.d.). QS world university rankings. Top Universities. https://www.topuniversities.com/world-university-rankings

scikit-learn developers. (n.d.). Cyclical feature engineering. scikit-learn. https://scikit-learn.org/stable/auto_examples/applications/plot_cyclical_feature_engineering.html

CollegeVine. (n.d.). Search cliff calculator. https://go.collegevine.com/search-cliff-calculator

American Association of Collegiate Registrars and Admissions Officers. (n.d.). China's new privacy law, U.S. CUI regulations spark confusion. https://www.aacrao.org/advocacy/compliance/china%27s-personal-information-protection-law-pipl/china%27s-new-privacy-law-u.s.-cui-regulations-spark-confusion

PRS Legislative Research. (n.d.). Digital Personal Data Protection Bill, 2023. https://prsindia.org/billtrack/digital-personal-data-protection-bill-2023

Towards Data Science. (n.d.). Dealing with features that have high cardinality. https://towardsdatascience.com/dealing-with-features-that-have-high-cardinality-1c9212d7ff1b

Data Head Hunters. (n.d.). Gradient boosting vs AdaBoost: Battle of the algorithms. https://dataheadhunters.com/academy/gradient-boosting-vs-adaboost-battle-of-the-algorithms/

Ibeta, P. (n.d.). Imbalanced learning. GitHub. https://pibieta.github.io/imbalanced_learning/notebooks/pablo-baseline-experiment.html

Let's Data Science. (n.d.). Frequency encoding. https://letsdatascience.com/frequency-encoding/

KDnuggets. (2022). Confusion matrix, precision & recall explained. https://www.kdnuggets.com/2022/11/confusion-matrix-precision-recall-explained.html

KC, S. (n.d.). Handling imbalanced dataset in machine learning. Medium. https://shivamkc01.medium.com/handling-imbalanced-dataset-in-machine-learning-9ac075787e07