Gateway to Opportunity: An Overview of Non-Immigrant Visas

Kartik Thakkar
Department of Computer Science
Utah State University
Logan, Utah
thakkarkartik30@gmail.com

***Abstract*—This paper provides an in-depth analysis of H-1B and F1 visa data, aiming to develop a predictive model for H-1B visa application outcomes. Starting with a comprehensive dataset of 116,000 records and over 150 features, we used data pre-processing, feature engineering, and sampling techniques aimed to handle data imbalance and improve model performance. Our strategy involved reducing feature dimensionality, inputting missing values, encoding categorical variables, and consolidating visa statuses into two categories for binary classification.**

We explored the effectiveness of the Random Forest Classifier under various configurations and sampling approaches to deal with the large imbalance between ’certified’ and ’denied’ visa statuses. Experiments included natural data imbalance, manual up/down-sampling, and synthetic data augmentation using SMOTE. Our results indicated that synthetic balancing significantly improved the model’s ability to accurately predict outcomes. This suggested that generating new samples is more effective than traditional resampling methods.

I. Introduction

A. H1B and F1 Visa Data and Pre-processing

Starting with F1 data, our data was found on travel.state.gov and we compiled Q4 Issuance data from the 2023 Fiscal Year and visa wait time data. On top of that we scraped data from U.S. News that gave us data on universities with the most International Students that use a form of F1 visa with python’s import of BeautifulSoup.

B. H1B and F1 Visa Analysis

H-1B Visa is a visa under the Immigration and Nationality Act and enables U.S. employees to employ workers from foreign countries. A F1 Visa is a student visa which allows a student to temporarily live in the United States for a period time while they study. These types of non-immigrant visas were particularly interesting to us since we had F1 visa students during the analysis and future H1B visa applicants as they prepare to look for sponsorship from companies.

The following figure (1) shows the top 10 countries from the past 21 years of international students.

The graph shows the increase in demand of applicants for the F1 visa within the past couple years with the only real outlier being the COVID years.

Taking a look at figure 2 we can take a look at that we found as we analyzed our data was whee the total issuance’s came from which lined up well with the top 10 countries, but more specifically we were able to see the cities where there were U.S. Posts for them to apply for the F1 Visas.

As expected most of the highest ones came from the capitals of the corresponding countries.

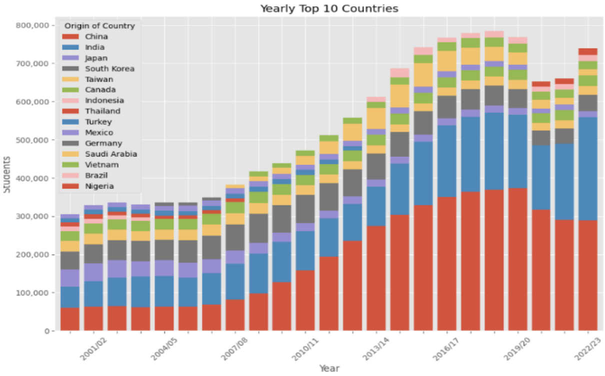

Fig. 1. Top 10 Countries with international students

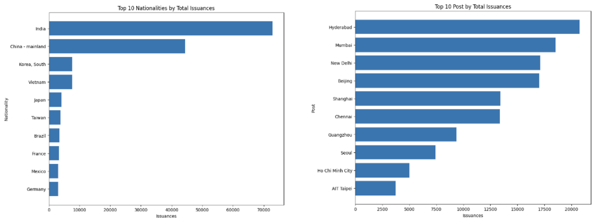

Fig. 2. Nationality and Post of Issuance

Our next figure (figure 3) will show the Yearly Academic Field Popularity and see how much it has changed within the two time frames as well seeing a really big increase in Math and Computer Science. With this data we are able to understand F1 demographic better and see what kinds of students are coming into the states for the specific professions in hopes of finding a correlation between that and the sponsors for their H1B future.

For H1B data, we also found it on travel.state.gov and did it the same way but filtered for H1B instead of F1. We also found Performance data from the U.S. Department of Labor which gives us a better insight on applications and approval ratings.

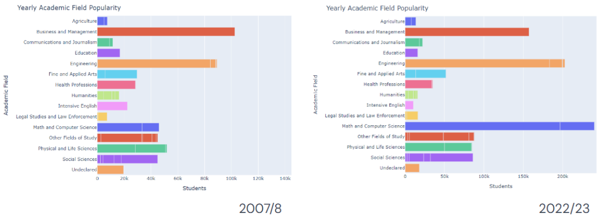

Fig. 3. Yearly Academic Field Popularity

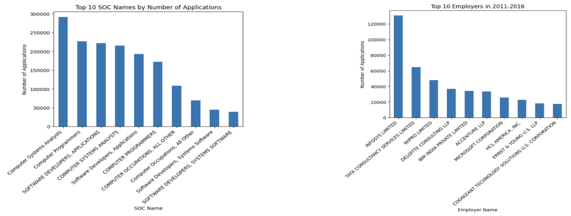

Fig. 4. F1 Visa Categorization

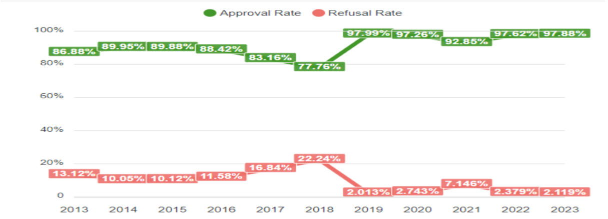

Fig. 5. Approval Rates

We can quickly see the correlation between what we found in the F1 Visa results from students translate into the companies that are the main sponsor as well as the line of work with the H1B data we collected looking at figure 4.

We can also see the percentage of approval rates increasing as demand for these jobs increase in the states in figure 5.

Now it's time for us to find out what factors can help predict visa approvals or rejections.

C. Predicting H-1B Visa Status

After doing analysis on both types of Visas we decided to go through and work on being able to create a model that can predict the status of an H1B Visa application. For this project, the goal was to achieve the following:
"Given a set of industrial, academic, employer-specific and demographic features, predict whether or not an H-1B visa will be certified or denied."

D. Dataset - an overview

The original raw H1B Status dataset was a high-dimension, high-imbalance dataset with 116,000 records and 150+ features. These features were from varying areas. For example, employer-specific features like company size, company establishment year, company location, etc. Employee demographic features like highest level of education, university, country of residence, etc. Job-specific features like required major, required training, experience, number of positions, etc. Other miscellaneous features included if the employer had a representative attorney, wage-related features, and employee education-related features.

E. Data Pre-processing

First, feature dimensionality was reduced from 150+ to 18. These 18 features were manually selected based on relevance to this project and ease of understanding. It can also be said that these features were sampled in a way and all the experiments done were done using these sampled features. Some of these sampled features included: minimum experience, candidate country of citizenship, candidate major, candidate's state of employment, employer state, employer size, employer's year of commencement, etc.

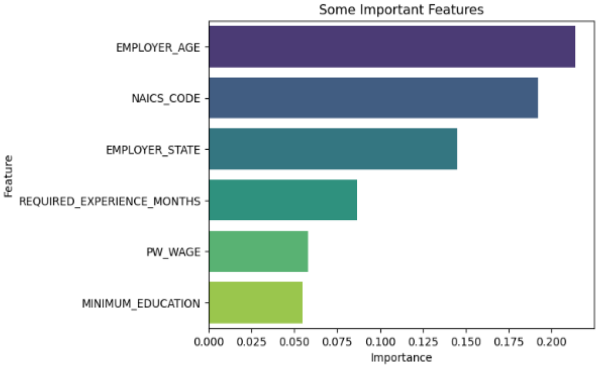

Fig. 6. Important Features of our Data

Second, null or missing values for instances where a value of 0 could mean a true zero value meaning that a 0 value would imply the absence of that particular value, 0 value imputation was done. For example, in case of null or missing values for the features `required_experience` and `required_experience`, it was assumed that a 0 value would mean no experience required and no training required respectively. Additionally, other records with null and missing values for features other than those discussed above were dropped.

Third, continuous numeric features with huge value ranges were essentially label encoded using regular mapping or based on manual and feature-specific binning criteria as required. This was done to make it easier for the classifier to find patterns or make decisions. For example, in the case of the prevailing wage feature `PW_WAGE`, where values ranged from a minimum of 25,000 to a maximum value greater than 3,000,000, the binning criteria were set as follows : (0,50000], (50000,75000], (75000,125000], (125000,150000], (150000,inf) and the following numeric labels were assigned : [0,1,2,3,4]. Another such feature is the number of employees for an employer `EMPLOYER_NUM_EMPLOYEES`, the huge range of values for this feature comes from the observation that not all employer has a huge employing power but only some do, which is why, maximum values tend to act like outliers. For this feature we binned the values into four labels [0,1,2,3]. The bins were as follows: (0,5000], (5000,25000], (25000,80000], $(80000,\infty)$.

F. Feature engineering and cleaning

It is often seen in natural data, that for some features, values can take up many variations but mean the same thing. These variations could be of many varieties, especially in strings, and can come from many sources like differences in the case for a string, misspelling of a string by a user at the time of data collection or by data handler, the addition of extra characters to a string, etc. To deal with this problem and to make data as consistent as possible, fuzzy string matching can be used. It is a handy tool that essentially finds similarities between two strings based on how much they are alike. This is calculated based on a threshold for similarity. A Python module named fuzzywuzzy provides this functionality. We utilized fuzzy string matching in the following ways to make our data more consistent.

The `'EMPLOYER_STATE'` feature had variations in spellings and characters. For instance, a record would have 'California' as a value for the employer state, but there would also be variations for the same value like: 'California', 'CALIFORNIA', 'California

34523', etc. To deal with this, we created a list of all US and used it against all values in this feature to match using fuzzy string matching. Another feature, `'MAJOR_MATCH'` was engineered using 2 features: `'MAJOR_FIELD_OF_STUDY`, a feature about employers' and `'FOREIGN_WORKER_INFO_MAJOR'`, a feature about employees. This was done to further reduce the dimension of input data. This engineered feature indicates whether or not the candidate has an educational background in the required major. This was done using fuzzy string matching where for each record, the corresponding column values were fuzzy string matched. A similarity threshold of 80% was used.

One drawback of this method was it did not work for cases where the two values were logically the same or implied the same professional domain but the words were different. For example, 'digital media' and 'graphic design' belong to the same line of work but have different spellings.

G. Feature Encoding

The `'EMPLOYER_STATE'` feature was one hot encoded. We believed that since a visa is a matter where the country of citizenship of the candidate is of great importance, the `'COUNTRY_OF_CITIZENSHIP'` feature was also one hot encoded despite having a large number of unique values.

All the yes/no features like `'REFILE'`, `'REQUIRED_TRAINING'`, `'REQUIRED_EXPERIENCE'`, `'FOREIGN_WORKER_CURR_EMPLOYED'`, `'MAJOR_MATCH'` were also processed to have 0/1 values.

H. Changing the target feature to have 2 categories

Originally, the `'CASE_STATUS'` feature which indicates the status of the H1-B visa for each record, had 4 distinct classes: Certified, Certified - Expired, Withdrawn, and Denied. The value counts for these classes were 42097, 41069, 5066, and 2132. Due to this imbalance and for ease of understanding, we changed this so that we work on a binary classification problem and put Certified, Certified - Expired, and Withdrawn into the class 'certified' and Denied into the class 'denied'. These two classes were eventually converted to 0 and 1.

Finally, after cleaning and processing the data to make it easy to use, the dimensions were 90,000 records and 186 features. It is to be noted that the features related to employer state and candidate country of citizenship were one hot encoded. Originally, without one hot encoding, around 15 features were present.

II. Method

For this binary classification problem, we used the Random Forest Classifier because of its ability to work with both categorical and continuous variables, not being affected by outliers as much, and most importantly - having access to feature importance so that feature selection can be carried out as needed. Both these qualities - having categorical and continuous variables as well as having outliers were characteristics of the data we had.

With the cleaned data we prepared, the huge class imbalance between certified (2132) and denied (88232), we experimented with various sampling techniques by keeping the classifier type (Random Forest) the same for each experiment.

III. Experimental Results

- Experiment 1 - **Random Forest Classifier and naturally imbalanced data**: We used a random forest classifier with default parameters with the original imbalanced data with a train-test split of 70-30. The results were skewed towards the majority class and performed perfectly for it whereas the performance was very poor for the minority class.
- Experiment 2 - **Random Forest Classifier and naturally imbalanced data with important features**: The important features were extracted from the random forest classifier from experiment 1 with a threshold for the importance of 0.025 to check performance with a better feature selection. The results were very similar to the experiment with no improvements and the same poor performance for the minority class.
- Experiment 3 - **Random Forest Classifier and balanced training data: minority class upsampled in training**: With the same train-test split of 70-30, the minority class in the training set was upsampled using the resample method from sci-kit learn. There was no improvement in performance than the last two experiments and the results were still skewed towards the majority class. The classifier failed at identifying and correctly predicting the minority class instances.
- Experiment 4 - **Random Forest Classifier and balanced training data: majority class downsampled in training**: At this time, the majority class in the training set was downsampled in the training as opposed to upsampling the minority class in experiment 3. There were significant improvements in the classifier's ability to identify minority class instances as indicated by high recall value and it was better than all classifiers from previous experiments but the classifier badly struggled to correctly predict the minority class instances.
- Experiment 5 - **Balanced Random Forest Classifier and Imbalanced Data**: As quoted from the official documentation for the imbalanced-learn module in the balanced random forest classifier section, *"A balanced random forest differs from a classical random forest by the fact that it will draw a bootstrap sample from the minority class and sample with replacement the same number of samples from the majority class"*, we used a balanced random forest classifier instead of a random forest classifier with our original imbalanced data.
- Experiment 6 - **Balanced Random Forest Classifier and Imbalanced Data with important features**: Feature importance from experiment 5 was used to filter the features with a threshold importance value of 0.025. A balanced random forest classifier was trained on important features.

The performance from both the classifiers (Experiment 5 and 6) was similar to Experiment 4. Both the classifiers struggled to correctly predict minority class instances. For both balanced random forest classifiers, a max depth of 40 and 100 number of estimators were used as parameters.

- Experiment 7 - **Random Forest Classifier and Balanced Data**: Pandas sampling functionality was used to perfectly balance the data before splitting it into training and testing. This classifier was equally good at identifying and correctly predicting both minority and majority classes.
- Experiment 8 - **Random Forest Classifier and balancing**

data using SMOTE : Synthetic Minority Oversampling Technique or SMOTE is an effective data augmentation technique that helps in balancing the data by generating samples for the minority class to match the number of samples of the majority class underneath; it uses distances and clusters to do this. We used a variation of SMOTE named SMOTENC which does the same job but also works with categorical columns. The dataset was balanced using SMOTENC and a random forest classifier was trained on it. The best results were achieved from this experiment. The classifier was great and correctly identified and predicted both minority and majority classes.

IV. Observations and Comparisons

An overall comparison can be done among all the experiments we tried mainly in terms of how well they identify and correctly predict instances from both minority and majority classes. The goal was to train a binary classifier that effectively identifies and predicts maximum samples from both classes.

For experiments 1 and 2, where imbalanced data was used, the classifier was highly biased towards the majority class, this aligns with the conventional wisdom that highly imbalanced data results in bad performance. It was observed that in both experiments, the recall and precision values were nearly perfect for the majority class and were the opposite for minority class instances.

For other experiments, different sampling strategies were used. For example, in experiments 3 and 4, the resample functionality was used from the sci-kit learn package to over-sample the minority class and under-sample the majority class in the training dataset. It was observed that up-sampling the minority class did not give any improvements but down-sampling the majority class improved the classifier's ability to identify instances from the negative class.

For experiment 7, where whole data was balanced and then split into training and testing sets, the classifier performed significantly better than any of the classifiers in the previous experiments. Although the precision and recall values were not very high for either of the classes, they were all greater than 0.70.

For experiment 8, synthetic oversampling for the minority class was done, the best results were achieved with this. All four precision and recall values were very high.

Experiment	Resampling	Precision		Recall		Overall Accuracy
		Class 0	Class 1	Class 0	Class 1	
1	None	0.39	0.98	0.07	1.00	0.98
2	None	0.46	0.98	0.06	1.00	0.98
3	sklearn Upsampling Minority	0.10	0.98	0.17	0.96	0.95
4	sklearn Downsampling Majority	0.06	0.99	0.73	0.71	0.71
5	None	0.06	0.99	0.75	0.72	0.72
6	None	0.06	0.99	0.75	0.71	0.71
7	pandas Downsampling Majority	0.74	0.74	0.76	0.72	0.74
8	imblearn SMOTENC	0.97	0.97	0.97	0.97	0.97

TABLE I
EXPERIMENTAL EVALUATION METRIC VALUES

V. Conclusion

When the dataset was manually balanced by sampling the same number of instances from the majority class that matched the total instances from the minority class, before splitting into training and testing sets, the classifier performed much better than all other experiments where some kind of sampling strategy was used to balance the data. Other experiments where down-sampling and up-sampling were used did very poorly in the case of minority class and especially in correctly predicting the instances from minority class. However, when SMOTE was used as a method to augment new data for the minority class and balance the data, the performance metric values skyrocketed and gave the best results. A reason for this could be attributed to the fact that when up-sampling is done using traditional methods, the data is essentially duplicated from the existing data, which does not add any variance to the data for the classifier to learn from. Similarly, when down-sampling is done based on a single feature, records are removed thereby causing a loss of information. SMOTE, on the other hand, does not randomly duplicate or remove any data, instead, it synthesizes new data from existing data belonging to the minority class that are similar to data points in the minority class but not duplicates. Therefore, it can be said that the best method for the goal of predicting the case status for an H1-B visa would be to have a balanced and as much data as possible, but in case there is a huge class imbalance, using synthetic samples for minority class to balance the dataset would be the best way to go.

VI. Future work

Some of the capabilities and functionalities we wish to achieve include more meaningful features like the ruling party at the time of visa issuance, monthly rolling rejection and approval rates by the visa issuing authorities and further integrating the impacts of geopolitical relations of the United States of America with other countries in the world.

VII. References

1) **Monthly Non-immigrant Visa Issuance Statistics:**
https://travel.state.gov/content/travel/en/legal/visa-law0/visa-statistics/nonimmigrant-visa-statistics/monthly-nonimmigrant-visa-issuances.html
2) **US Visa Approval, Refusal Rates by Visa Type - 2024, History:**
https://visagrader.com/visa-approvals-and-refusals
3) **Global Visa Wait Times:**
https://travel.state.gov/content/travel/en/us-visas/visa-information-resources/global-visa-wait-times.html
4) **Prediction/Clustering Model:**
https://www.dol.gov/agencies/eta/foreign-labor/performance
5) **2024 Universities with the Most International Students — US News Rankings:**
https://www.usnews.com/best-colleges/rankings/national-universities/most-international