Katrina Apiado
Zoe Atkins
Sharmaine Parani

Predicting the Likelihood of Obtaining An H1-B Visa

Abstract

For nonimmigrant workers seeking to reside in the United States temporarily, applying to the H1-B program is a complex process. This project will create a model that predicts the likelihood of foreign workers obtaining an H1-B visa, in which applications were filed by the employer on the workers' behalf. The dataset used for training and testing has 35 input variables, with the case status of either "CERTIFIED" or "DENIED" as the output. Our team will be meeting at least once a week to stay on course to the completion of the project. Throughout the duration of the project, our team will train and verify the effectiveness of the model and present the results with a presentation and final report.

Problem Description

"More than 100,000 legal immigrants — 28 percent of the family-sponsored and employment-based lines with quotas — waited a decade or more to apply for a green card in 2018, up from 3 percent in 1991" [1]. Currently, the demand for the application process for a green card has increased over the years, but the supply for green cards have been restricted by a need for a sponsor and the H1-B visa lottery system, where they only take 20,000 out of the 65,000 capped applicants [2]. The 20,000 selected are not guaranteed a visa as they undergo another process that checks eligibility and visa requirements where 32% are only accepted, which further restricts applicants in a time-consuming process. For this project, the main goal is to analyze the likelihood of an international applicant in receiving an H1-B visa based on descriptive features like country of birth, employer, etc.

In order to see the success rate of obtaining a visa, the objective of this project will be to create a model that will be able to predict what critically affects the application process for an H1-B visa based on different types of features.

Dataset Summary

The dataset we chose to use is named "H1-B FY2018" for the fiscal year of 2018 and can be found under LCA Programs (H-1B, H1-B1, E-3) [3]. This dataset was chosen as it would be representative of present-day circumstances due to its general nature pre–COVID-19. Out of the 52 descriptive features, 27 were applicable to our model. The 27 descriptive features include original dataset features and features added to the dataset from original columns. The following field names and their respective descriptions are given from the U.S. Department of Labor [3].

- **CASE_STATUS** (output) = Status associated with the last significant event or decision. Valid values include "Certified" and "Denied." Excluded values from the model are "Certified-Withdrawn" and "Withdrawn."
- **CASE_SUBMITTED_MONTH** = Month the application was submitted.

- **EMPLOYMENT_START_MONTH** = Beginning month of employment.
- **EMPLOYER_STATE** = State information of the Employer requesting temporary labor certification.
- **EMPLOYER_POSTAL_CODE** = Postal code information of the Employer requesting temporary labor certification.
- **EMPLOYER_REGION** = Region information of the Employer requesting temporary labor certification.
- **AGENT_REPRESENTING_EMPLOYER** = Y = Employer is represented by an Agent or Attorney; N = Employer is not represented by an Agent or Attorney.
- **AGENT_ATTORNEY_STATE** = State information for the Agent or Attorney filing an H-1B application on behalf of the employer.
- **SOC_CODE2** = First two digits of occupational code associated with the job being requested for temporary labor condition, as classified by the Standard Occupational Classification (SOC) System.
- **SOC_CODE4** = Last four digits of occupational code associated with the job being requested for temporary labor condition, as classified by the Standard Occupational Classification (SOC) System.
- **NAICS_CODE3** = Three digits; Industry code associated with the employer requesting permanent labor condition, as classified by the North American Industrial Classification System (NAICS).
- **TOTAL_WORKERS** = Total number of foreign workers requested by the Employer(s).
- **NEW_EMPLOYMENT** = Indicates requested worker(s) will begin employment for new employer, as defined by USCIS I-29.
- **CONTINUED_EMPLOYMENT** = Indicates requested worker(s) will be continuing employment with same employer, as defined by USCIS I-29.
- **CHANGE_PREVIOUS_EMPLOYMENT** = Indicates requested worker(s) will be continuing employment with same employer without material change to job duties, as defined by USCIS I-29.
- **NEW_CONCURRENT_EMP** = Indicates requested worker(s) will begin employment with additional employer, as defined by USCIS I-29.
- **CHANGE_EMPLOYER** = Indicates requested worker(s) will begin employment for new employer, using the same classification currently held, as defined by USCIS I-29.
- **AMENDED_PETITION** = Indicates requested worker(s) will be continuing employment with same employer with material change to job duties, as defined by USCIS I-29.
- **FULL_TIME_POSITION** = Y = Full Time Position; N = Part Time Position.
- **WAGE_RATE_OF_PAY_FROM** = Employer's proposed wage rate.
- **WAGE_UNIT_OF_PAY** = Unit of pay. Valid values include "Hour", "Week", "Bi-Weekly", "Month", or "Year".
- **H1B_DEPENDENT** = Y = Employer is H-1B Dependent; N = Employer is not H-1B Dependent.
- **WILLFUL_VIOLATOR** = Y = Employer has been previously found to be a Willful Violator; N = Employer has not been considered a Willful Violator.
- **SUPPORT_H1B** = Y = Employer will use the temporary labor condition application only to support H-1B petitions or extensions of status of exempt H-1B worker(s); N =

Employer will not use the temporary labor condition application to support H-1B petitions or extensions of status for exempt H-1B worker(s);

- **LABOR_CON_AGREE** = Y = Employer agrees to the responses to the Labor Condition Statements as in the subsection; N = Employer does not agree to the responses to the Labor Conditions Statements in the subsection.
- **WORKSITE_STATE** = State information of the foreign worker's intended area of employment.
- **WORKSITE_POSTAL_CODE** = Zip Code information of the foreign worker's intended area of employment.
- **WORKSITE_REGION** = Region information of the foreign worker's intended area of employment.

Data Cleaning and Preprocessing

Due to the large nature of the dataset, Jupyter Notebook was used to begin the cleaning process of the data. The data itself was unclean and had random inputs alongside null values. We dropped null values found in columns that we found significant.

Initially, we weren't able to get dummy variables and this was due to the large number of categorical variables found in WORKSITE_STATE and EMPLOYER_STATE. To fix this, label encoding was implemented to convert this object type to integer type.

With consultation from the professor, new columns were made from the original columns in the dataset: CASE_SUBMITTED_MONTH, EMPLOYMENT_START_MONTH, WORKSITE_REGION, EMPLOYER_REGION, SOC_CODE2, SOC_CODE4, NAICS_CODE3.

- CASE_SUBMITTED_MONTH is originally from the column CASE_SUBMITTED and EMPLOYMENT_START_MONTH is originally from the column EMPLOYMENT_START_DATE, in which only the month was extracted from these dates.
- EMPLOYER_REGION and WORKSITE_REGION were formed by taking the EMPLOYER_STATE and WORKSITE_STATE columns and categorizing the states into "West," "Midwest," "Northeast," and "South."
- SOC_CODE2 and SOC_CODE4 were formed by taking the SOC_CODE column and separating the code by its hyphen delimiter. SOC_CODE was first converted to a string, and then converted to integer once in their two columns.
- NAICS_CODE3 is originally from NAICS_CODE, where the first three characters of the string were kept and then converted to integer.

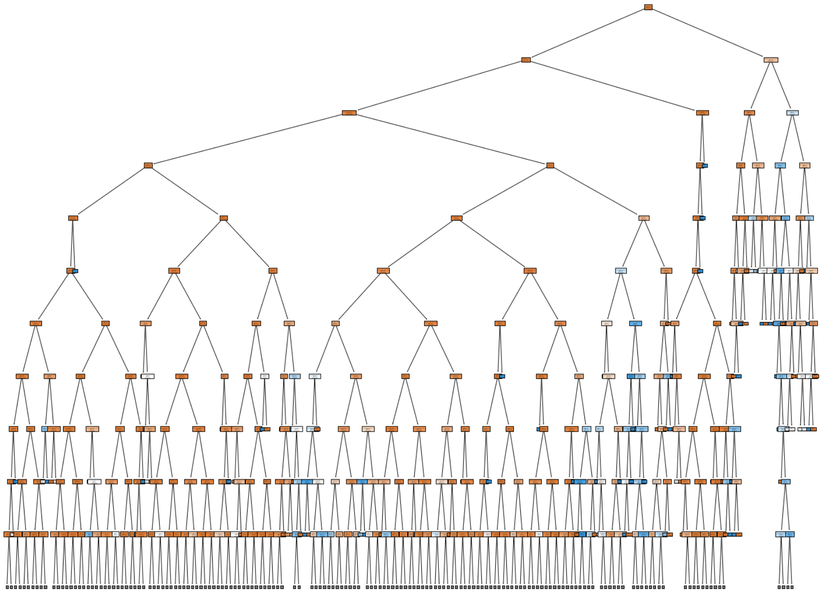
After cleaning the entire dataset, 573,512 out of 654,360 rows were able to be retained. The dataset proved to be highly imbalanced with 566,066 rows with the outcome of "CERTIFIED" and the remaining 7,446 rows with the outcome of "DENIED." Due to the data being highly imbalanced, stratified k-fold was implemented to take into account this imbalance.

Methodology

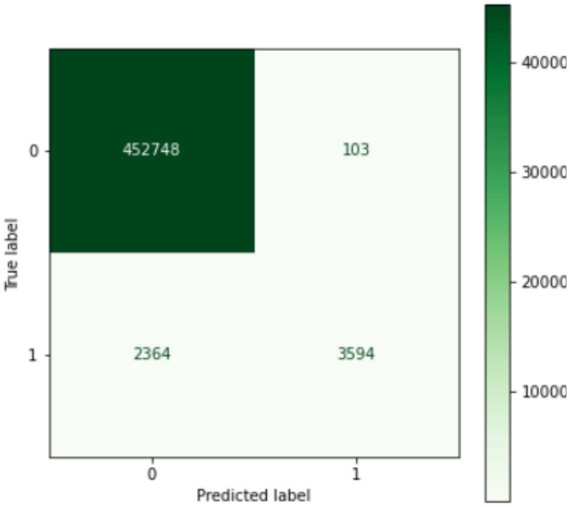
As this contained an outcome variable with a historical dataset, this was categorized as supervised learning. Additionally, this dataset was identified as a classification problem, in which we implemented the following classification models: Decision Tree, Random Forest, and AdaBoost. After cleaning the data, we imported the modeling packages for scikit machine learning. We tested and analyzed each of the models with machine learning techniques and evaluated it with performance metrics.

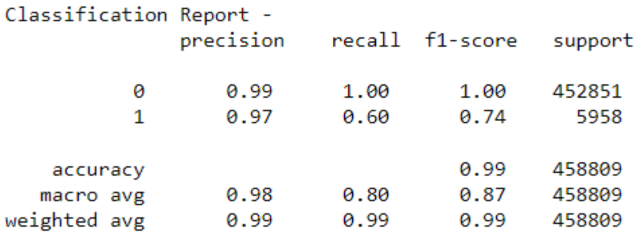
I. Decision Tree Model

For the Decision Tree model, we made sure to convert all the input features into integers or floats by creating dummy variables for categorical features. From this dataframe, we partitioned the data into a training and testing set, where test size in this case was 0.2. We fit the training set onto the decision tree model and got an initial decision tree, as seen below. Some features to note would be that the max depth was 10 rather than "none" since the initial model run with "none" never completely finished running, even after letting the model run for several hours. Thus, the max depth was constrained to 10 for the means of a faster run.

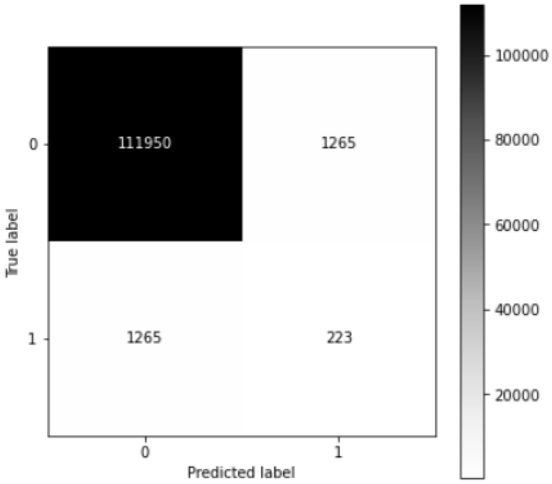


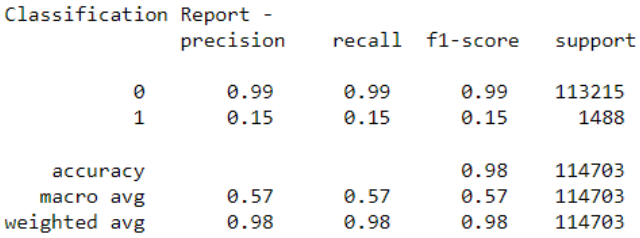
From this initial decision tree, we analyzed its performance based on a confusion matrix and classification report. Since we partitioned the data, we wanted to see the performance of the training set on the training data as well as the testing set on the testing data. Usually, an ideal training set in a confusion matrix would be near perfect as most of the values would be accurately predicted for the outcome classes, which are "CERTIFIED" and "DENIED". However, as seen in the confusion matrix in green below, values for the "CERTIFIED" were more accurately predicted than it was for "DENIED". This is most likely due to the highly imbalanced dataset that we are utilizing. Similarly, the classification report for training data performance has a macro average F1-score of 0.87, which is not ideal but may be considered good performance for a highly complex dataset.




```
Classification Report -
               precision    recall  f1-score   support

           0       0.99      1.00      1.00    452851
           1       0.97      0.60      0.74      5958

    accuracy                           0.99    458809
   macro avg       0.98      0.80      0.87    458809
weighted avg       0.99      0.99      0.99    458809
```

We also tested the testing set on the training data and analyzed its performance accordingly with a confusion matrix and classification report, as seen below. The confusion matrix for the testing set and training set show similar matrices, where "CERTIFIED" outcomes were predicted accurately while "DENIED" outcomes were predicted inaccurately. The classification report shows a 30% decrease from the training set, where the testing set's average macro F-1 score was 0.57. This would mean that the testing set is being overfitted.

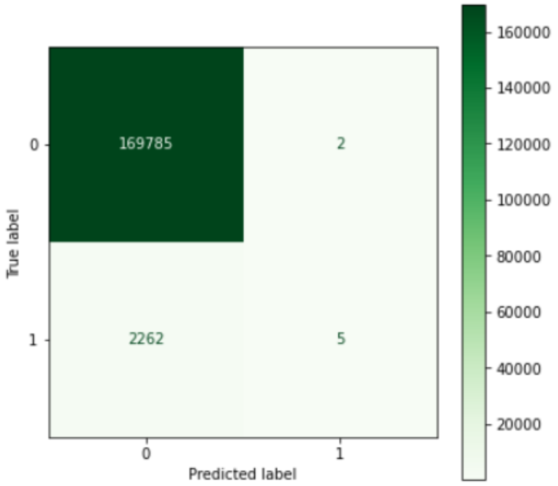


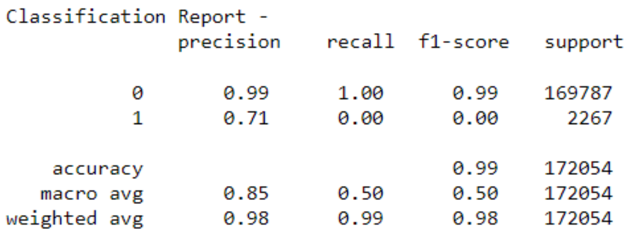

```
Classification Report -
               precision    recall  f1-score   support

           0       0.99      0.99      0.99    113215
           1       0.15      0.15      0.15      1488

    accuracy                           0.98    114703
   macro avg       0.57      0.57      0.57    114703
weighted avg       0.98      0.98      0.98    114703
```

From here, we wanted to see if we could obtain a better performance by hyperparameter tuning the testing set. We made a list of initial guesses and adapted the hyperparameters accordingly. From there, we tested the adapted hyperparameters with the testing set to determine if hyperparameter tuning has a major effect on performance improvement. We called a grid search cross validation, where we implemented 5 stratified folds, utilized F1-macro scoring, and had the parameter grid as the hyperparameters. Then, we created a classification report and confusion matrix for the set after hyperparameter tuning and only noticed a small difference. One thing to note specifically is that the testing set before and after hyperparameter tuning have the same macro F-1 score of 0.57. This illustrates that there may have been slight changes, but performance of the testing set may have only improved by less than one percent. Thus, even though there was room to continue hyperparameter tuning, we decided to stop tuning when considering the excess amount of time it took to fit the model into randomized search cross-validation, as well as the little performance improvement it illustrated from both the confusion matrix and classification report. Due to the long amount of time it took to get one model finished, one thing to note from here on forward is that we reduced the amount of stratified k-folds to 3 folds as well as reduced the number of iterations to 10.

II. AdaBoost Classification Model

After importing packages and the dataset, we determined which features are predictors and which would be the outcome for the model. Then, the data is partitioned into training and testing sets before defining the AdaBoost model. An initial guess for the parameters, which are the learning rate and the number of estimators, were set and placed into a random grid. We called and fitted the model into randomized cross validation (CV), where we implemented 3 stratified folds and utilized F1-macro scoring. From this fitted model, the initial random CV score is 0.498 and initial random CV parameters would be 225 for the number of estimators and 1.80 for the learning rate. The random grid helped guide us to the best improved CV score and best CV parameters which could be integrated when fitting the model into grid search CV. From this fitted model, we analyzed the confusion matrix and classification report and found the following:



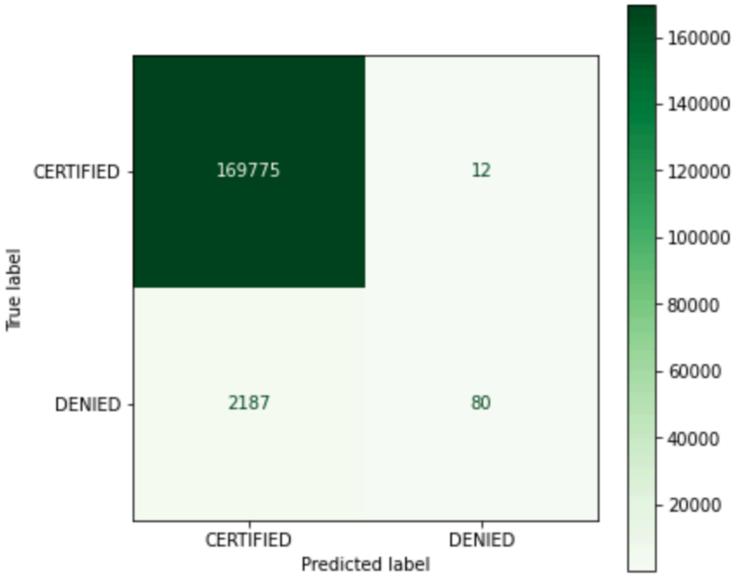


```
Classification Report -
              precision    recall  f1-score   support

           0       0.99      1.00      0.99    169787
           1       0.71      0.00      0.00      2267

    accuracy                           0.99    172054
   macro avg       0.85      0.50      0.50    172054
weighted avg       0.98      0.99      0.98    172054
```

As seen in the confusion matrix above, the confusion matrix is not able to accurately predict “DENIED” from the model but is more likely able to predict “CERTIFIED” correctly. Furthermore, the classification report gives an average macro F-1 score of 0.50, most likely due to the near-perfect F-1 score for “CERTIFIED” and a score of zero for “DENIED”. Like the decision model, this may be due to the highly imbalanced data which skews the model more towards one outcome than the other.

III. Random Forest Classification Model

For our last model, we use the Random Forest classification model. We went through a similar process as the last two models where we start by importing the data and splitting 20% of the data into a testing group and the other 80% into the training data. The training data was then used to train the model, and the fitted model was tested on the testing data. After the trial and error of using multiple k-folds and iterations in previous models, we settled on 3 folds and 10 iterations to fit the data. The confusion matrix below shows how the model performed on the testing data.



Below is the classification report of how the model performed on the testing data. The most notable scores are that the macro average f1-score is 0.53 and the weighted average f1-score is 0.98. These scores signify that the model is just as good as the other two in predicting a certified outcome. In terms of predicting the denied outcome for applicants, the model correctly predicts when the applicant is denied better than the AdaBoost model, but not as well as the Decision Tree model.

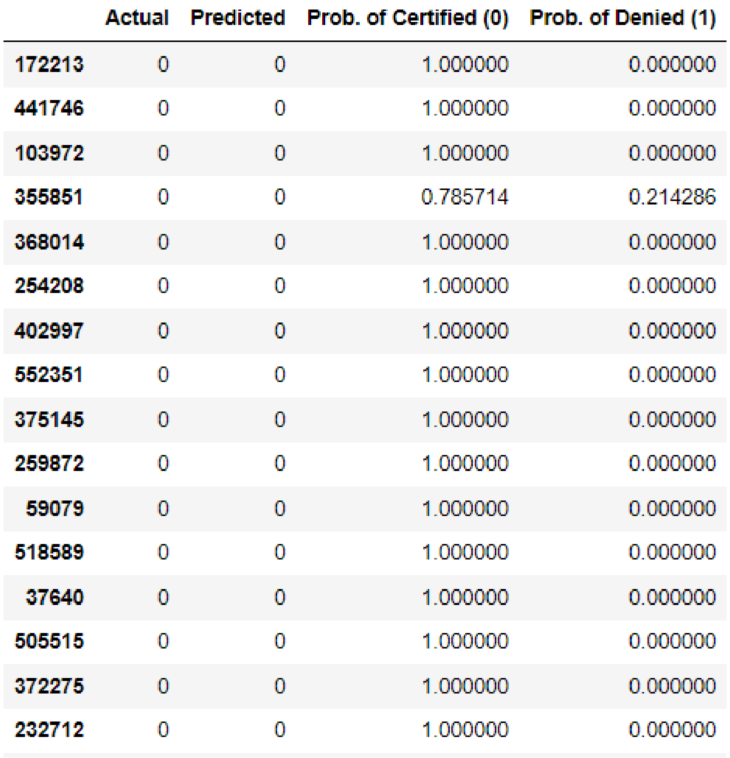
```
Classification Report -
               precision    recall  f1-score   support

   CERTIFIED       0.99      1.00      0.99    169787
      DENIED       0.87      0.04      0.07      2267

    accuracy                           0.99    172054
   macro avg       0.93      0.52      0.53    172054
weighted avg       0.99      0.99      0.98    172054
```

Performance Evaluation

After analyzing the performance of each model, there were further performance metrics to evaluate the model, like the predictive performance and the feature importance. Here, we will evaluate each of the model’s performance metrics and compare similarities and differences between them.

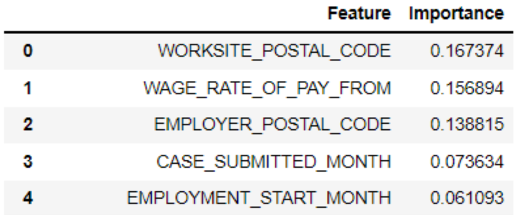
I. Decision Tree Model

When looking at the decision tree's predictive performance from below, we can see that the actual and predicted outcome is 0, or in this case, would be "CERTIFIED" while 1 would be "DENIED". The probability for certified for the first few features in the table are 1 or near 1 and based on this pattern, one of the analyses that we made is that the certified outcome is not only more likely to be predicted, but it will more likely be the result in this case. This may also be due to the fact that the dataset is highly imbalanced to begin with.

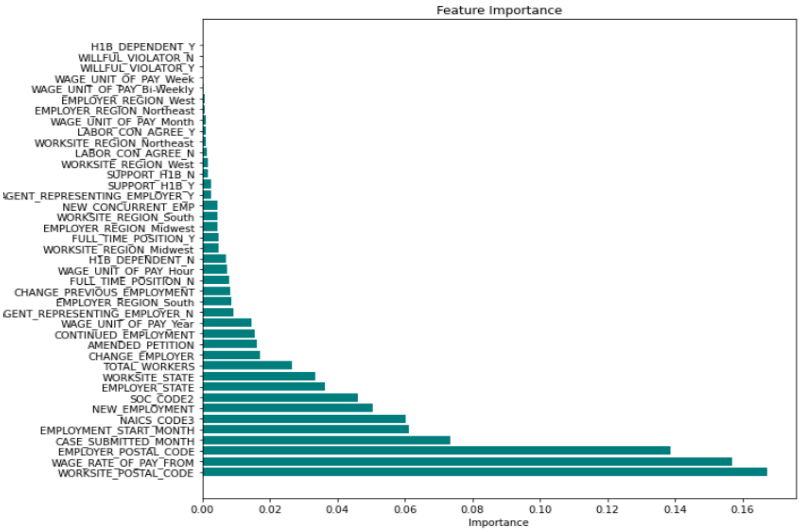


	Actual	Predicted	Prob. of Certified (0)	Prob. of Denied (1)
172213	0	0	1.000000	0.000000
441746	0	0	1.000000	0.000000
103972	0	0	1.000000	0.000000
355851	0	0	0.785714	0.214286
368014	0	0	1.000000	0.000000
254208	0	0	1.000000	0.000000
402997	0	0	1.000000	0.000000
552351	0	0	1.000000	0.000000
375145	0	0	1.000000	0.000000
259872	0	0	1.000000	0.000000
59079	0	0	1.000000	0.000000
518589	0	0	1.000000	0.000000
37640	0	0	1.000000	0.000000
505515	0	0	1.000000	0.000000
372275	0	0	1.000000	0.000000
232712	0	0	1.000000	0.000000

Additionally, we analyzed the top 5 important features that are most likely to influence the outcome variable, and the top 5 for this model would be WORKSITE_POSTAL_CODE, WAGE_RATE_PAY_FROM, EMPLOYER_POSTAL_CODE, CASE_SUBMITTED_MONTH, and EMPLOYMENT_START_MONTH.

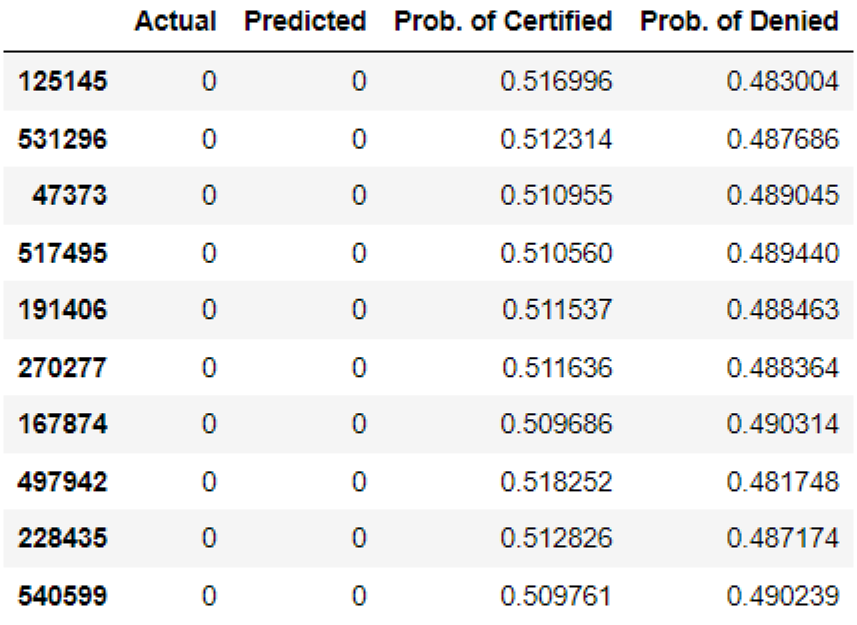


	Feature	Importance
0	WORKSITE_POSTAL_CODE	0.167374
1	WAGE_RATE_OF_PAY_FROM	0.156894
2	EMPLOYER_POSTAL_CODE	0.138815
3	CASE_SUBMITTED_MONTH	0.073634
4	EMPLOYMENT_START_MONTH	0.061093



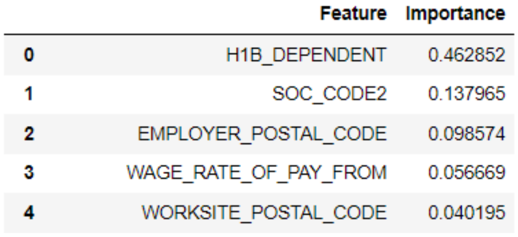
II. AdaBoost Classification Model

Like the decision tree model, the predictive performance and feature importance were also found. As seen in the results below, the predictive performance illustrates that the probability between certified and denied was about even. From this, we can give a general deduction that certified was chosen over denied by a small percentage for most of the dataset, which is most likely the reason why denied was rarely predicted, as seen in the confusion matrix for this model.

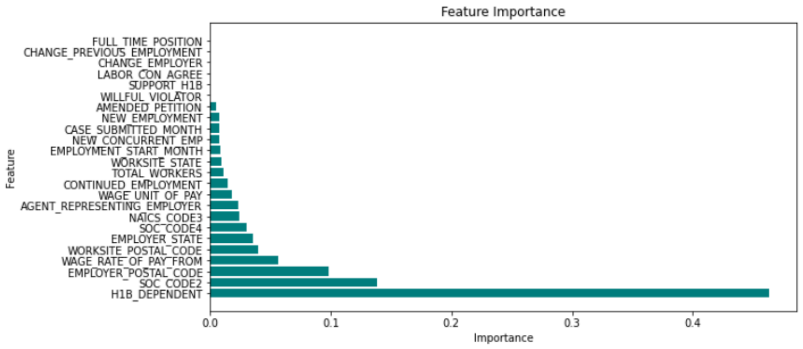


	Actual	Predicted	Prob. of Certified	Prob. of Denied
125145	0	0	0.516996	0.483004
531296	0	0	0.512314	0.487686
47373	0	0	0.510955	0.489045
517495	0	0	0.510560	0.489440
191406	0	0	0.511537	0.488463
270277	0	0	0.511636	0.488364
167874	0	0	0.509686	0.490314
497942	0	0	0.518252	0.481748
228435	0	0	0.512826	0.487174
540599	0	0	0.509761	0.490239

Meanwhile, as seen below, the top 5 features for this model would be H1B_DEPENDENT, SOC_CODE2, EMPLOYER_POSTAL_CODE, WAGE_RATE_OF_PAY_FROM, and WORKSITE_POSTAL_CODE. One interesting thing to note here would be the feature importance for H1B_DEPENDENT with an importance of 46% while the other four remaining candidate feature importance percentages sum up to 33.4% altogether. In short, this model gave one input feature more importance than the rest, which may have affected its overall performance in some aspects.

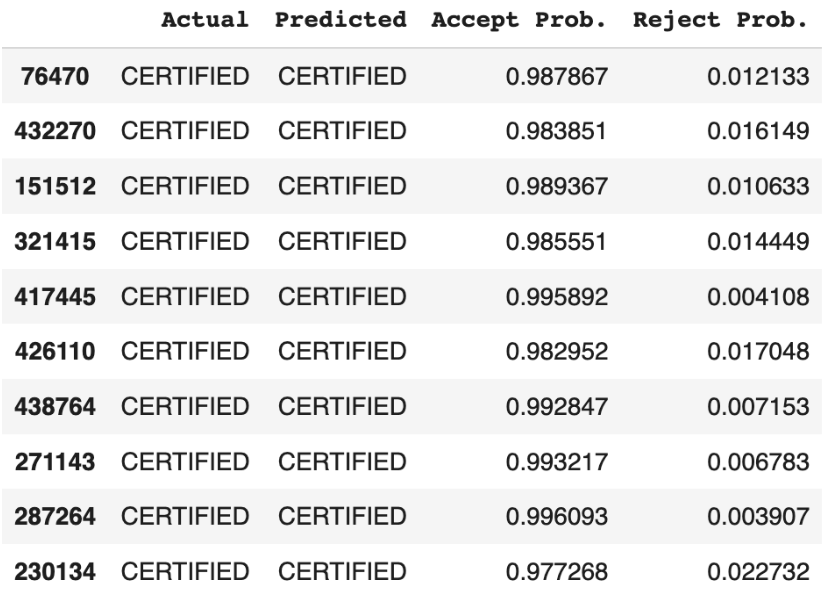


	Feature	Importance
0	H1B_DEPENDENT	0.462852
1	SOC_CODE2	0.137965
2	EMPLOYER_POSTAL_CODE	0.098574
3	WAGE_RATE_OF_PAY_FROM	0.056669
4	WORKSITE_POSTAL_CODE	0.040195



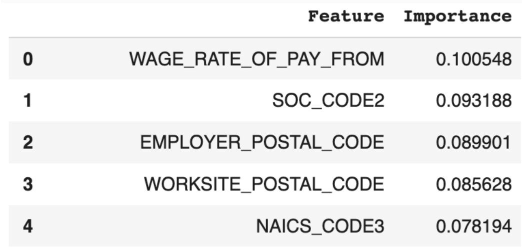
III. Random Forest Classification Model

Along with the AdaBoost and Decision Tree models, we checked the predictive performance and the feature predictions for the Random Forest model. Below is a glimpse into the predictive performance of how the model predicted whether the applicant was certified or denied. Here, you'll notice that under the "Actual" and "Predicted" columns, the results are not labeled with 1's and 0's like the previous two models. Instead they are labeled with the name of the outcome, "CERTIFIED" or "DENIED", because this model did not require the data to be LabelEncoded, which means the data was never converted to ones and zeros. Taking a look at the accepted and rejected probabilities show that the model was very certain in its prediction, unlike the previous model that had both scores relatively equal close.

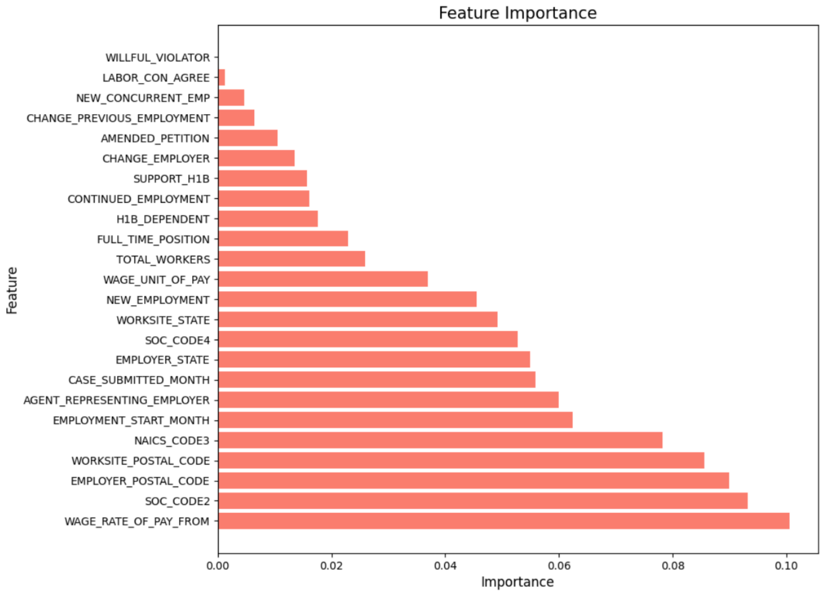


	Actual	Predicted	Accept Prob.	Reject Prob.
76470	CERTIFIED	CERTIFIED	0.987867	0.012133
432270	CERTIFIED	CERTIFIED	0.983851	0.016149
151512	CERTIFIED	CERTIFIED	0.989367	0.010633
321415	CERTIFIED	CERTIFIED	0.985551	0.014449
417445	CERTIFIED	CERTIFIED	0.995892	0.004108
426110	CERTIFIED	CERTIFIED	0.982952	0.017048
438764	CERTIFIED	CERTIFIED	0.992847	0.007153
271143	CERTIFIED	CERTIFIED	0.993217	0.006783
287264	CERTIFIED	CERTIFIED	0.996093	0.003907
230134	CERTIFIED	CERTIFIED	0.977268	0.022732

When taking a look at the feature importance, the top five features are: WAGE_RATE_OF_PAY_FROM, SOC_CODE2, EMPLOYER_POSTAL_CODE, WORKSITE_POSTAL_CODE, and NAICS_CODE3. The graph of feature vs. importance shows how all the features compare against each other. Compared to the previous AdaBoost model, the importance of the features are much more balanced.



	Feature	Importance
0	WAGE_RATE_OF_PAY_FROM	0.100548
1	SOC_CODE2	0.093188
2	EMPLOYER_POSTAL_CODE	0.089901
3	WORKSITE_POSTAL_CODE	0.085628
4	NAICS_CODE3	0.078194



Results

Between the three models, the Decision Tree model performed the best with a macro average F1-score of 0.57, compared to the AdaBoost model's score of 0.50 and the Random Forest model's score of 0.53. Considering that all three had the same weighted average F1-score of 0.98, this signifies that the difference between the models' macro average F1-score was their ability to accurately predict which applicants were denied an H-1B visa. With this in mind, the Decision Tree model is the best performer at predicting both whether an applicant was denied or certified an H-1B visa.

Knowing the Decision Tree model predicted the most accurately, the feature importance from that model is the most notable with the top five features being, as mentioned previously: WORKSITE_POSTAL_CODE, WAGE_RATE_PAY_FROM, EMPLOYER_POSTAL_CODE, CASE_SUBMITTED_MONTH, and EMPLOYMENT_START_MONTH.

Assumptions

We assumed that all H-1B visa petitions were submitted and verified by the U.S. Department of Labor.

Ethical Considerations

For our ethical considerations, we kept in mind how difficult and stressful of a process it is for applicants to be awarded an H-1B visa with the lottery system. With this project, we were not intentionally finding which features were the best predictors for whether an applicant was awarded an H-1B visa or not, but instead whether there was a pattern or bias to the lottery system in place.

Limitations

Our greatest limitation was how large our dataset was at over 500,000 rows. This created issues when we tried fitting the models with a larger number of folds and iterations. Additionally, the dataset was largely imbalanced, which led to the models struggling to accurately predict when an applicant was denied a visa. This issue was prevalent in the AdaBoost model which ended up predicting that all the applicants were certified a visa, since they were more likely to be certified than denied, according to our imbalanced dataset.

In the future, we would want to include the other types of visas: E-3 Australian, H-1B1 Chile, and H-1B1 Singapore. These visas were excluded from our dataset due to their low volume and to keep our models consistent, but we would like to analyze these visas within a more balanced dataset.

Conclusions

As American citizens, we often overlook how we are born in the United States and immediately given citizenship. There are so many people who apply for H-1B visas every day and it is up to a lottery system on whether they can stay in the country.

After analyzing the data, we could see that even the most important features did not have a weighted impact of more than 0.17 on whether the application was certified or denied. This is ultimately good news because it confirms that the process is randomized and not biased.

Statement of Individual Contributions of Team Members

Katrina Apiado:
I was responsible for finding the data source and cleaning the dataset.

Zoe Atkins:
I was responsible for the Random Forest model, while assisting Sharmaine with the other models.

Sharmaine Parani:
I was responsible for running Decision Tree and AdaBoost models with the cleaned dataset.

All three of us contributed to the report and final presentation for this project.

References

Bier, D. (2019, June 18). *Immigration Wait Times from Quotas have Doubled: Green Card Backlogs are Long, Growing, and Inequitable*. Papers.ssrn.com. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3501812

H-1B Specialty Occupations, DOD Cooperative Research and Development Project Workers, and Fashion Models | USCIS. (2021, February 5). Www.uscis.gov. https://www.uscis.gov/working-in-the-united-states/h-1b-specialty-occupations

Performance Data | U.S. Department of Labor. (n.d.). Www.dol.gov. https://www.dol.gov/agencies/eta/foreign-labor/performance

Upadhye, N. (2021, April 1). *H-1B Visa Lottery (How Does it Work?)*. NNU Immigration. https://www.nnuimmigration.com/h1b-visa-lottery/