Predicting Case Status of H-1B Visa Petitions

Induja Sreekanthan
(A53206643, isreekan@ucsd.edu)

Jahnavi Singhal
(A53205623, jsinghal@ucsd.edu)

Prahal Arora
(A53219500, prarora@ucsd.edu)

Rahul Dubey
(A53070477, rvdubey@ucsd.edu)

***Abstract*— H-1B visa class is among the most sought after visa-categories. The applications for H-1B visa differ in many ways: the company name, prevailing wages, workplace location, the type of job, job title, year of petition and alike. In this report, we attempt to predict the status of the visa petition based on the visa-application metadata. The intent of this study is to discover how visa status outcome is influenced by attributes of user application. The classifier designed in this report could be utilized by both, H-1B aspirants and employers, to gauge the likelihood of visa certification, before and after filing the petition.**

***Index Terms*— H-1B, visa, status, prediction, classification, regression, model, analysis**

I. INTRODUCTION

The H-1B is an employment-based, non-immigrant visa category for temporary foreign workers in the United States. For a foreign national to apply for H1-B visa, a US employer must offer a job and petition for H-1B visa with the US immigration department. This is the most common visa status applied for and held by international students once they complete college / higher education (Masters, PhD) and work in a full-time position. H-1B visa class is very industry relevant and many individuals and companies rely heavily on this yearly allotment. Currently theres no way for knowing how application attributes contribute towards the final visa-status outcome.

In this report, we proceed in the following manner to achieve our objective. In section II, we introduce the dataset that we experimented on and our analysis on it. Then we talk about the details of our predictive task in section III. Section IV describes the various features we extracted from the data based on our analysis. Then, we present the related work that has been done on H1-B visa classification. Section VI discusses the models in details, their advantages and the issues we faced. In the last section, we present our results and conclusion.

II. DATA SET

We downloaded a dataset about H1-B visa petitions from Kaggle [1]. It consists of more than 3 million data points. The data labels are divided into 4 classes: (1) Certified (2) Denied (3) Withdrawn (4) Certified-Withdrawn.

We have the following information for each sample:

- **Name of the employer**: Name of employer submitting labor condition application.
- **Year of filing petition**: Year in which the H-1B visa petition was filed
- **SOC name**: Occupational name associated with the SOC_CODE which is an occupational code associated with the job being requested for temporary labor condition, as classified by the Standard Occupational Classification (SOC) System.
- **Job title**: Title of the job
- **Wage**: Prevailing Wage for the job being requested for temporary labor condition. The wage is listed at annual scale in USD. The prevailing wage for a job position is defined as the average wage paid to similarly employed workers in the requested occupation in the area of intended employment. The prevailing wage is based on the employers minimum requirements for the position.
- **Full-time position**: Y = Full Time Position; N = Part Time Position
- **Work site**: City and State information of the applicant's intended area of employment in USA

TABLE I
DISTRIBUTION OF LABELS

Label	Number of samples
Certified	2615623
Denied	93567
Withdrawn	89799
Certified-Withdrawn	202659

As can be seen from the table above, this is an

imbalanced data set. The certified samples are much more in number than the denied ones. For this reason, preprocessing of the data is one of the most important step of our work.

A. *Preprocessing & Pruning*

The number of samples which we considered for the experimentation was 200,000 (after preprocessing and filtering from 3 million samples). Firstly, we removed all the data points which had ”N/A” or empty values for case status, employer name, year, full-time position and worksite. Worksites are given in the form ”City, State”. We extracted the states from worksite.

In our models, we only included the cases ’CERTIFIED’ and ’DENIED’ and these were labeled ’1’ and ’0’ respectively. We decided to ignore ’CERTIFIED-WITHDRAWN’ and ’WITHDRAWN’ since those were decisions taken by the applicant and/or employer. To get a balanced data of size 200,000, we retained all the data labeled 0 and a random sample of positively labeled data of size 100,000.

B. *Exploratory Analysis*

We analyzed the distribution of various features, their relation with the number of applications, and with the acceptance rate.

1) Acceptance rate vs Year: We define Acceptance rate as a fraction of applications that received ’CERTIFIED’ status.
We analyzed the correlation of the acceptance rate with the year of application and we see that year can be a good feature in determining the case-status.

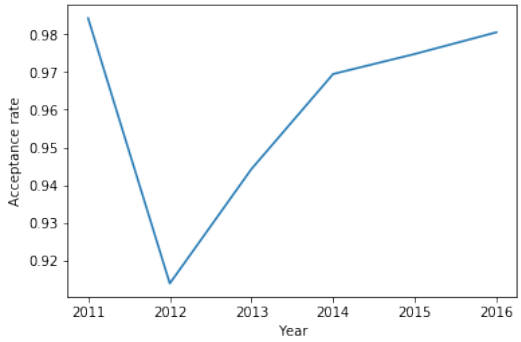

Fig. 1. Year of application v/s acceptance rate

2) Median wage vs Year: We analyzed the variation of the median wage with year, and we can see that the median wage of H1-B applicants increases almost linearly with year.

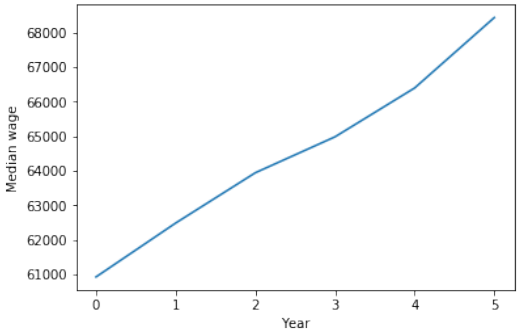

Fig. 2. Year of application v/s median wage

3) Employer-name: We explored the data for different employers who file H1-B petitions. Below listed are top 10 employers who file the most number of petitions:

TABLE II
TOP 10 EMPLOYERS WITH MOST NUMBER OF APPLICATIONS

EMPLOYERS	**No. of applications**
Infosys Limited	130241
Tata Consultancy Services Limited	64346
Wipro Limited	43645
Deloitte Consulting LLP	36667
Accenture LLP	32983
IBM India Private Limited	28164
Microsoft Corporation	22373
HCL America, Inc.	22330
Ernst & Young U.S. LLP	18213
Larsen & Tourbo Infotech Limited	16724

The word cloud below shows the popular employers in the visa applications.

Fig. 3. Word cloud of Employer names

It is interesting to observe that maximum number of visa applications are filed by Indian IT giants such as Infosys, Tata Consultancy Services(TCS) and Wipro.

We have 45482 unique Employer-names in our dataset. We sorted the different employers according to

their acceptance rates. From Figure 4 we see that employer vs acceptance rate is not a uniform distribution. And hence, it can be a good predictor of case-status in our prediction task.

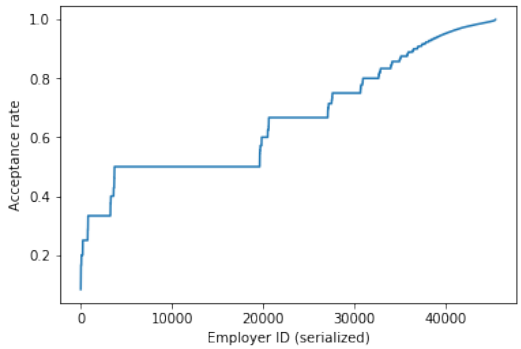

Fig. 4. Employers v/s Acceptance rate

4) SOC-Name: SOC-Name is nothing but the occupational name associated with the job being requested for temporary labor condition, as classified by the Standard Occupational Classification (SOC) System. Below we list the SOC-names which have the highest acceptance rate:

TABLE III
TOP 10 SOC-NAME WITH HIGHEST ACCEPTANCE RATE

SOC-NAME	Acceptance rate
Computer Systems Analyst	0.99936
Software Quality Assurance Engineers and Testers	0.99903
Computer Systems Engineers	0.99871
I.T. Project Manager	0.99865
Computer Systems Architect	0.99796
Business Intelligence Analysts	0.99655
Computer Programmer	0.99605
Network and Systems Administrators	0.99476
Mathematicians	0.99427
Software Developers	0.99300

The word cloud below shows the popular Job titles in the visa applications.

Fig. 5. Word cloud of Job titles in visa applications

This word cloud is in-line with our intuition. Various Information Technology (IT) roles dominate the H-1B visa applications.

Now, we have 1228 unique SOC-names. We sorted the SOC names according to their acceptance rates. Thus, we can see from Figure 6 that this is not a uniform distribution, and hence SOC-NAME would be a good feature in our predictive task.

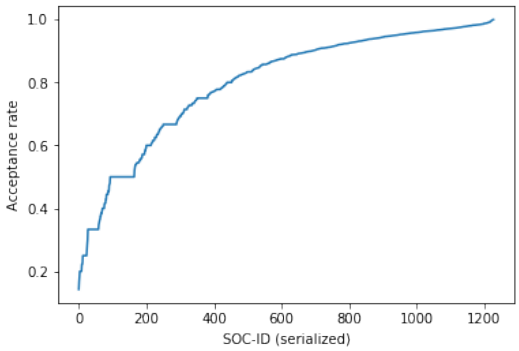

Fig. 6. Occupation codes v/s Acceptance rate

5) State / Location: Using heatmap, we have visualized the distribution of total number of visa applications across states. Color intensity shows the variation. Large number of applications is indicated by a darker shade.

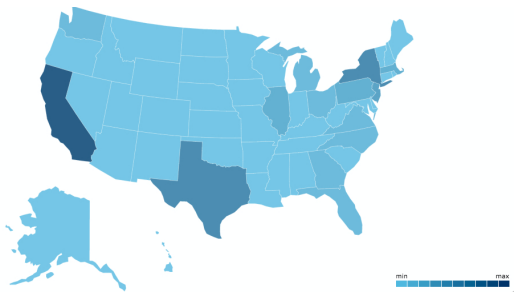

Fig. 7. State-wise visa applications numbers

We also visualized the median wage across states. The heatmap below shows the distribution of the median wage where states with darker shade attribute to more wages compared to others.

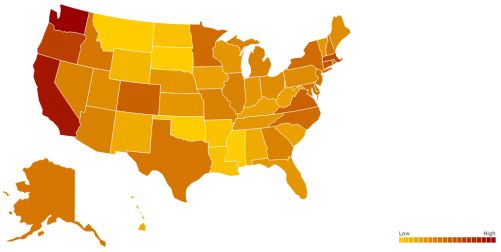

Fig. 8. State-wise Median wage

From the above drawn heatmaps, we infer that a large number of applications are sent from California and New York. Moreover, we also see an interesting wage distribution across the states. For example, wage on the west coast is much higher than any other state. Wage on the east coast is higher than the states in middle region. Due to the presence of such variation, we decided to use 'State' as one of our features.

Thus, by doing this exploratory data analysis, we have been able to extract features that are relevant to this predictive task.

III. PREDICTIVE TASK

In this report, we endeavor to study correlation between visa application attributes and the final status outcome. We perform a classification task, which takes in the information about applicants' prevailing wage, employer name, state the worksite belongs to, occupation code, full-time status, year of petition and predicts the visa status as 'DENIED' or 'CERTIFIED'.

To set a benchmark for our classification task, we adopted the following two baselines -

- **ZeroR (Baseline 1)**: The simplest classification method which relies on the target and ignores all predictors. ZeroR classifier simply predicts the majority category (class).
- **Biased random sampling (Baseline 2)**: This classification method predicts the outcome for a given sample on the basis of a biased coin toss. These biased probabilities are proportional to the number of occurrences of each class (Certified / Denied).

To evaluate the performance of classifiers against the baselines described above, we use following metrics of evaluation:

- **Classification Accuracy**: Measure of proximity of measurement results to the true value.

$$acc = \frac{tp+tn}{tp+tn+fp+fn}$$

where 'tp' is the number of true positives,
'tn' stands for true negative,
'fp' stands for false positive and
'fn' is false negative.

- **False Positive Rate**: Refers to the probability of falsely rejecting the null hypothesis for a particular test.

$$FPR = \frac{fp}{fp+tn}$$

- **False Negative Rate**: The proportion of positives which yield negative test outcomes with the test, i.e., the conditional probability of a negative test result given that the condition being looked for is present.

$$FNR = \frac{fn}{fn+tp}$$

- **Balanced Error Rate**: The measure of the proportion of wrong classifications in each class.

$$BER = \frac{1}{2} * (FNR + FPR)$$

- **F1 Score**: This measures accuracy using precision p and recall r.
- Precision is the ratio of true positives (tp) to all predicted positives (tp + fp).
- Recall is the ratio of true positives (tp) to all actual positives (tp + fn).

$$\text{F1 score} = \frac{2*precision*recall}{precision+recall}$$

IV. FEATURE SELECTION

We experimented with a set of features and this is the list of features we found to be most useful for our prediction task.

- **Year**: one-hot encoding of 'year of application'. This adds a dimension of 6.
- **State**: one-hot-k vector to represent the states. This adds a dimension of 52 (one each for the 50 States, plus District of Columbia and Puerto Rico).
- **Prevailing Wage**: We have split the prevailing wages into buckets as given in the table below.

TABLE IV

WAGES SPLIT INTO BUCKETS

Wage split (1000$)	No. of samples
0 - 40	23654
40 - 50	22757
50 - 55	18378
55 - 60	20981
60 - 65	21445
65 - 70	16273
70 - 80	24781
80 - 90	16167
90 - 110	17759
110 - 150	9712
150 - 200	2874
more than 200	3279

- **Employer-name**: We have a large number of unique values for employer names. One-hot encoding will put the curse of dimensionality on our predictive task, since we have 45482 unique employers. Instead of encoding each employer with a one-hot-k vector, we added
  - Number of applications by employer
  - Acceptance rate of employer

  We found that by encoding our employers using these features, our feature dimensions remained feasible, and also provided us with a good predictive accuracy.
- **SOC-name**: We have 1228 unique SOC names. We represented the SOC names in our feature vector in the same way as we did for employer name. Instead of adding each SOC name with a one-hot-k representation vector, we added
  - Number of applications with the SOC-NAME
  - Acceptance rate of applications with the SOC-NAME
- **Full-time**: This feature takes in 1 / 0 as yes / no for full time position

V. LITERATURE SURVEY

The dataset that we are studying is available on Kaggle (a platform for data science competitions) under the name 'H-1B Visa Petitions 2011-2016 dataset' . This is processed from from the original data available on Office of Foreign Labor Certification (OFLC) website, which albeit clean, is not suitable for rapid analysis. The contributor has performed a series of data transformations making the data more accessible for exploration.

The Office of Foreign Labor Certification (OFLC) generates program data that is essential both for internal assessment of program effectiveness and for providing the Department's external stakeholders with useful information about the immigration programs administered by OFLC. This page [2] includes program information organized in the form of quarterly and annual releases of program disclosure data to assist with external research and program evaluation. The webpage page allows the public to access the latest data in easily accessible formats for the purpose of performing in-depth longitudinal research and analysis. Office of Foreign Labor Certification (OFLC) case disclosure data is available for download by the federal fiscal year cycle covering the October 1 through September 30 period (all disclosure data sets are saved in the Microsoft Excel (.xls) file format).

From Data-Analytics point of view, this data is extensively studied, visualized and reported by the OFLC [2]. OFLC annually issues a report presenting employment-based cumulative immigration program data and analysis based on applications submitted to the Department by employers across the country. In addition to that, the OFLC has made available program fact-sheets, displaying key selected statistics about each of the major immigration programs. These efforts revolve around finding top Occupations, States, Employers and Industries that contribute to highest number of H-1B visa application. An independent exploratory data analysis on this dataset by Sharan Naribole[3] found that the Data Scientist position has experiences an exponential growth in terms of H-1B visa applications, and also that the Data Scientist jobs are clustered in a few hostpots such as San Francisco.

Our research tells us that a very few studies are conducted on this dataset which leverage Machine Learning models for prediction and exploratory analytics. An independent study by Andrew Shikiar [4] aims to predict wages of the visa recipients by performing text analysis of application attributes such as Job title, company name, employer name, visa status and work place city. They concluded that Occupational Classification and Job Title were the two most important fields to predict the applicants wage as accurately as possible. A project done by the students of UC Berkley [5] tried to predict the waiting time to get a work visa for a given job title and for a given employer. They used K-Nearest Neighbors as the primary model to predict 'Quickest Certification Rate' across both occupations and companies.

To the best of our knowledge, predicting the outcome of H-1B visa petition is an unexplored territory and our work would be first of its kind to make an attempt

to study the relationship between visa application attributes and the visa status.

VI. MODELS

We have used four different classifiers for the prediction task and evaluated the results of each.

A. Gaussian Naive Bayes

In Gaussian Naive Bayes, the model assumes that there exists a Gaussian distribution of features, and all features are independent of each other. The advantage of using Naive Bayes is that it is a simple and interpretable model, and converges quickly. The disadvantage is that it assumes that all our features are independent. We got a decent result with Naive Bayes, but not the best result.

B. Logistic Regression

In logistic regression, probability of the response taking a particular value is modeled based on combination of values taken by the predictors. Estimation in logistic regression chooses parameters that maximize the likelihood of observing the sample values. The advantage of Logistic Regression model is that it gives the confidence of prediction as a probability. The disadvantage is that it assumes that the classes are linearly separable in feature space.

C. Random Forest Classifier

Random Forest is an ensemble classifier that fits a number of meta-classifiers (here: Decision Tree), on various sub-samples of the dataset. It then averages the prediction which helps it improve the overall predictive power and control over-fitting. Random Forest has the advantage that it hardly overfits. But the disadvantage is that a lot of hyper parameters have to be tweaked which becomes time consuming.

D. AdaBoost Classifier

AdaBoost Classifier is a form of ensemble classifier that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases. This serves as a strong advantage in our case. The disadvantage of this is that it is slightly sensitive to noisy data.

VII. EVALUATION & RESULTS

Initially we used only 3 features: year, state, full-time. But our model was not performing well with any classifier, getting the maximum accuracy of 65%.

For some time, we worked with the whole data which was highly imbalanced (positive labels were 10 times more than negative samples). We used Logistic Regression and SVM Classifier, but both were giving poor results for the negatively labeled data. In order to give more weight to false positives compared to false negatives, we implemented Logistic Regression on tensor flow with the equation given below, where we included a multiplier of 10 to the negatively labeled data (Objective function's equation shown below). This also gave poor results and increasing the multiplier only resulted in divergence.

$$E(w) = -(\sum_{y_i=1} log(\sigma(X.\theta)) + 10 \sum_{y_i=0} log(1 - \sigma(X.\theta)))$$

We downsampled the data and worked on the rest of the experiments with 200,000 samples and included features for EMPLOYER_NAME, and SOC_NAME as one-hot-k representations. This largely increased the dimensionality of our model as the value of 'k' is large, which slowed down the performance.

After doing more analysis on the data, we came up with another idea which gave us the best results. Instead of encoding these categorical features as one-hot-k representations, we used the number of applications, and acceptance rate for each unique EMPLOYER_NAME, and SOC_NAME. Adding these 2 features **improved** our accuracy by approximately 20%.

After getting our final feature set, we tried several models as described in the previous section. Four different models, Gaussian Naive Bayes, Logistic Regression, Random Forest Classifier and AdaBoost Classifier, were successfully built using the best combination of the features. The data was split as **80:20** for training and testing.

The results of these models and some baseline models are presented below.

TABLE V

COMPARISON BETWEEN BASELINES

Baseline	F1 Score	FPR	FNR	BER	Accuracy
Baseline 1	1.0	0.68895	0.0	0.5	0.526
Baseline 2	0.525	0.5251	0.475	0.5	0.5

TABLE VI

COMPARISON BETWEEN VARIOUS MODELS

Models	F1 Score	FPR	FNR	BER	Accuracy
Naive Bayes	0.8075	0.1718	0.2177	0.1948	0.8041
Logistic Regression	0.8703	0.2157	0.0799	0.1474	0.8559
Random Forest	0.8756	0.2297	0.0610	0.1453	0.8592
AdaBoost Classifier	0.8756	0.2080	0.0766	0.1423	0.8613

For **optimizing** the hyper-parameters, we used Grid Search in the space of hyper-parameters, to find the hyper-parameters that give us the lowest error rate.

- Logistic Regression gave the best results with the inverse of regularization strength (C) equal to 2.

- Random Forest Classifier had the number of estimators (n_estimators) = 400 and max_depth = 15.

- AdaBoost had n_estimators = 100, base_estimator as Decision Tree and rest of the parameters were kept as default from scikit-learn.

Apart from these models, we also employed SVM classification for this task but, it **failed** to converge on our systems.

VIII. CONCLUSIONS

In this work, Gaussian Naive Bayes, Logistic Regression, Random Forest Classifier and AdaBoost Classifier were considered for determining the status of H1-B visa applications. AdaBoost Classifier performed the best in terms of accuracy and F1 score over others. We achieved a best of **86.139%** classification accuracy and Balanced Error Rate of **0.142** on validation data. We inferred that the state of worksite, year of application, prevailing wages, employer name and soc-name play an important role in determining the case status of an H-1B application. We observed that the most important feature to consider for our model is the acceptance ratio for the employer and the number of petitions filed by the employer. This clearly indicates the trends of H1-B visa filings which has a high correlation with the employer's acceptance rate.

REFERENCES

[1] https://www.kaggle.com/nsharan/h-1b-visa/version/2

[2] https://www.foreignlaborcert.doleta.gov/performancedata.cfm

[3] http://blog.nycdatascience.com/student-works/h-1b-visa-petitions-exploratory-data-analysis/

[4] https://blog.bigml.com/2013/10/01/using-text-analysis-to-predict-h1-b-wages/

[5] https://www.ischool.berkeley.edu/projects/2016/project-alien-worker