H-1B VISA PREDICTION MODEL

Under the Guidance of Dr. David Belanger

Created By:
Prathamesh Parchure
Graduate Student
Stevens Institute of Technology

Table of Contents

1. Introduction

The H-1B is an employment-based, nonimmigrant visa category for temporary foreign workers in the United States. For a foreign national to apply for H-1B 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. With tens of thousands of H-1B visa applications filed every year, we thought it would be useful for companies, job applicants and immigration agencies to build a model that accurately predicts the outcome of these petitions. Given the size of data for such a use case, there is a need to find the best machine learning algorithm on the basis of providing the best accuracy, as well as the one which takes the least amount of elapsed time. Therefore, in addition to building a predictive model, we also wanted to compare the performance of different machine learning algorithms when they are scaled in terms of dataset size and cluster size (i.e. number of nodes).

2. Platforms

Before proceeding with the model, research was done on few platforms that can be most suitable for the implementation. We also wanted to gain exposure to certain industry leading platforms and find a solution which would give us the flexibility to vary the cluster size. The following solutions were considered:

- **Databricks**

  It aims to help clients with cloud-based big data processing using Spark. Databricks develops a web-based platform for working with Spark, that provides automated cluster management and IPython-style notebooks.

- **Amazon Web Services (AWS)**

  It provides on-demand cloud computing platforms to individuals, companies and governments, on a paid subscription basis. The technology allows subscribers to have at their disposal a full-fledged virtual cluster of computers, available all the time, through the internet.

- **Cloudera**

  It provides a scalable, flexible, integrated platform that makes it easy to manage rapidly increasing volumes and varieties of enterprise data. It delivers the core elements of Hadoop - scalable storage and distributed computing – along with a Web-based user interface and vital enterprise capabilities.

- **Docker VM**

  Docker is a software technology providing containers, an additional layer of abstraction and automation of operating-system-level virtualization. It is flexible, lightweight and easy to create applications.

Databricks and AWS both looked suitable for our purpose but the cluster solution in both was a paid one so it was decided to use the university provided solution which was free. The development and scalability testing for increasing dataset sizes was done using a Docker image pre-installed with Spark and Jupyter notebook. The cluster based scalability testing was done on the university provided cluster.

3. Dataset

The dataset was downloaded from https://www.kaggle.com/nsharan/h-1b-visa. There were approximately 3 million visa applications from last 6 years i.e. from 2011 to 2016. There were 11 variables in total, 1 dependent and 10 independent. The table below describes these variables:

Variables	Description
Case_ID	ID of the petition filed.
Case_Status	Status associated with the decision. Valid values include “Certified,” “Certified-Withdrawn,” Denied,” and “Withdrawn”.
Employer_Name	Name of employer submitting labor condition application.
SOC_Name	It is the occupational code associated with the job being requested for temporary labor condition
Job_Title	Title of the job
Full_Time_Position	Y = Full Time Position; N = Part Time Position
Prevailing Wage	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.
Year	Year in which the H-1B visa petition was filed
Worksite	City and State information of the foreign worker's intended area of employment
lon	longitude of the Worksite
lat	latitude of the Worksite

4. Data Preparation

The following data cleaning and transformation tasks were completed to make the dataset ready for modeling:

- The dataset did not have a lot of missing values. There were missing values in Case Status, Worksite & Prevailing Wage columns. Rows with missing values in one or more of these columns formed less than 0.1% of the dataset so they were removed.
- The worksite column had the state and city name comma separated. It was split into two separate columns “State” and “City” to enable use of State as a predictor in the model.
- The longitude and latitude variables, which give information about worksite location, were removed as the city and state also give information about the worksite location. Creation of new features using latitude and longitude was considered but not implemented. This is discussed in the ‘Problems Encountered’ section.
- The SOC field describing the occupation had a lot of values, most of them similar to each other. Hence, using Python package called Collections, over 2000 SOC values were consolidated and separated into two categories: IT and non-IT.
- 99% of the case statuses comprised of ‘Certified’, ‘Certified Withdrawn’ and ‘Denied’. For the purpose of prediction, ‘Certified’ and ‘Certified Withdrawn’ were combined into ‘Certified’ category. This represented the successful

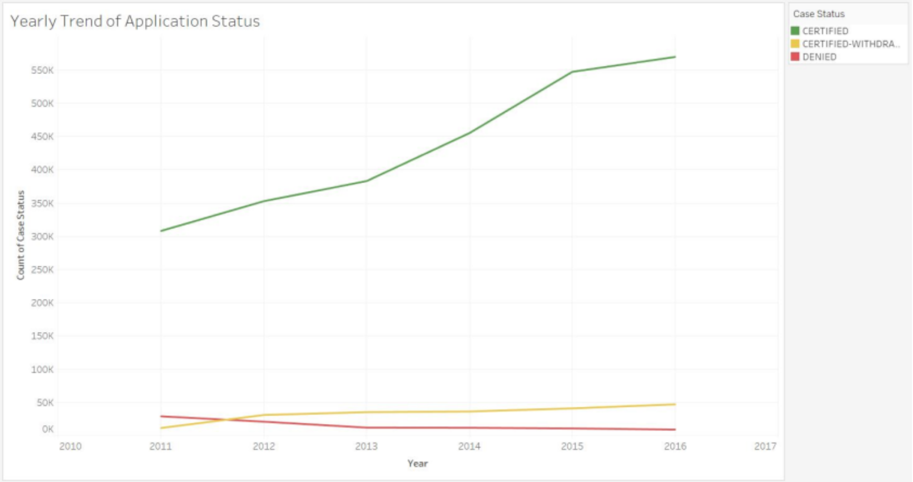
applications. ‘Denied’ represented the unsuccessful applications and the label of interest for prediction.

After the data cleaning and transformations, the number of rows reduced to approximately 2.9 million.

5. Data Exploration

Exploratory Data Analysis (EDA) was done to identify trends in the data and features that might be good predictors for the model.

The distribution of unsuccessful and successful applications showed an unbalanced dataset. ‘Denied’ applications formed only 3.3% of the dataset. The following graph shows the yearly distribution of successful and unsuccessful applications:



The unbalanced nature of the dataset is visible in the above graph. The number of successful applications (green line at the top) shows an upward trend historically while the number of unsuccessful applications (red line at the very bottom) shows a slight downward trend over the same period. A state wise analysis of visa applications was also conducted.

Top 5 states with highest denial rates:

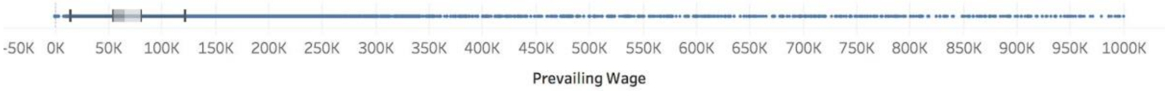
State	Denials as percentage of the total number of applications for the state
Puerto Rico	15.38%
Alaska	11.46%
Wyoming	11.26%
Hawaii	10.25%
Montana	8.47%

Top 5 states with lowest denial rates:

State	Denials as percentage of the total number of applications for the state
Washington	1.87%
Delaware	1.97%
Arkansas	2.06%
New Hampshire	2.13%
Wisconsin	2.15%

A comparison of the two SOC categories, IT and Non-IT, showed that 1.7% of IT applications and 4.4% of the non-IT applications were unsuccessful.

A boxplot of Prevailing Wage, shown below, shows that it is skewed to the right with more outliers present at the upper end. Values above $122000 are outliers and form about 3.8% of the dataset. These values were removed for modeling purpose.



6. Modeling

The dataset was split into a training and test set using a 70 / 30 split i.e. 70% of dataset was used for training and 30% was used for testing. Case Status was used as the binary target variable and SOC_Name, State, Full Time, Prevailing Wage were used as the predictors. The following three algorithms were used for model building:

- **Naive Bayes**

  Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Multinomial Naive Bayes classifier was used as one of the predictor features had more than two levels.

- **Binomial Logistic Regression**

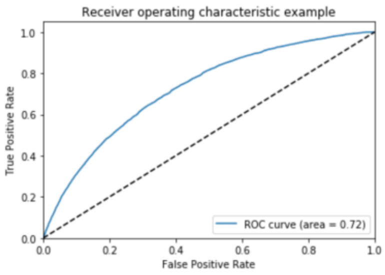
  Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. In Binomial Logistic Regression, the outcome is measured with a binary variable (in which there are only two possible outcomes).

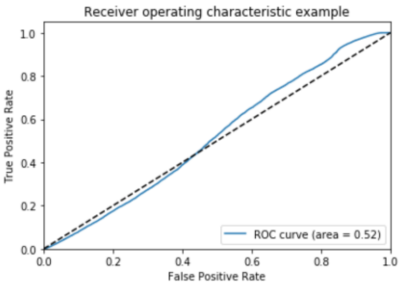
- **Random Forest**

  Random forest is an ensemble learning method for classification, regression and other tasks, that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. In this project, Random Forest was used for classification. Random Forest is also robust to data anomalies e.g. unbalanced data.

7. Metrics

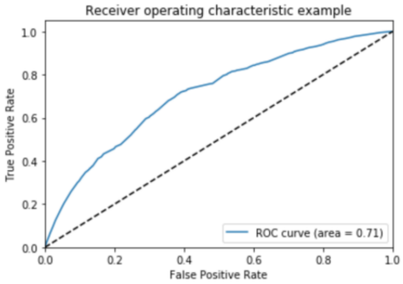
The following plots show the ROC curve and the Area Under the Curve (AUC) measure for each model for the full dataset.





Logistic Regression

Naive Bayes



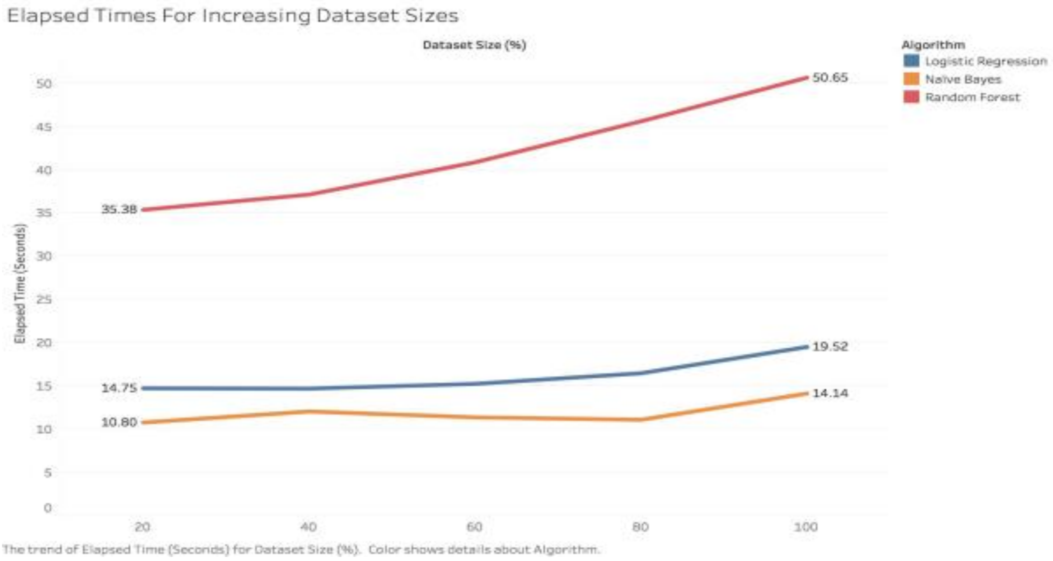
Random Forest

Considering the unbalanced nature of the dataset, Logistic Regression and Random Forest models are fair. Naive Bayes results in a very poor model.

In order to compare the model performances in terms of elapsed times, each model was run for 20%, 40%, 60%, 80% and 100% of the dataset. Following elapsed times (in seconds) were recorded for each run.

Dataset Size	**Elapsed Time**	**Algorithm**
20%	14.75	Logistic Regression
40%	14.71	Logistic Regression
60%	15.26	Logistic Regression
80%	16.49	Logistic Regression
100%	19.52	Logistic Regression
20%	10.8	Naive Bayes
40%	12.07	Naive Bayes
60%	11.39	Naive Bayes
80%	11.11	Naive Bayes
100%	14.14	Naive Bayes
20%	35.38	Random Forest
40%	37.14	Random Forest
60%	40.87	Random Forest
80%	45.61	Random Forest
100%	50.65	Random Forest

The following is a plot of the above times:



The trend of Elapsed Time (Seconds) for Dataset Size (%). Color shows details about Algorithm.

In the above plot, the red line at the very top represents Random Forest, blue line in the middle represents Logistic Regression and yellow line at the bottom represents Naive Bayes. Although fastest, Naive Bayes is unusable due to its lack of predictive capability. Logistic Regression and Random Forest both appear to be scalable models as they look linear. The AUC measure showed only minor change as dataset size was increased (around 0.01 increase).

Each of these models was also run on a 10-node cluster for the entire dataset and the elapsed times were recorded. The below table gives a comparison of the model elapsed times for 1 node versus 10-nodes.

Number of Nodes	**Elapsed Time**	**Algorithm**
1	19.52	Logistic Regression
1	14.14	Naive Bayes
1	50.65	Random Forest
10	19.11	Logistic Regression
10	11.62	Naive Bayes
10	44.52	Random Forest

Logistic Regression does not show any improvement when scaled in terms of parallelization but Naive Bayes and Random Forest do. Random Forest shows most improvement. This improvement can be attributed to parallel execution of Random Forest. It runs the ensemble of decision trees in parallel therefore an increase in number of nodes helps it to parallelize even more and finish faster.

8. Problems Encountered

The unbalanced nature of the dataset limited the predictive capability of the models. As ‘Denied’ applications only formed 3% of the dataset, the model did not have many examples to learn from. Lack of subject matter expertise with regard to SOC_Names was another problem faced during the data exploration and preparation stages. Access to such expertise could have resulted in different grouping of Standard Occupational Categories (SOC_Names) potentially increasing models’ predictive capabilities. The dataset does not have many features that can be used as predictors and it was difficult to add new features. The dataset includes Employer Names which is not useful as a predictor however industry information of these employers could have been used as a predictor feature. Our research did not reveal any straightforward way of obtaining this information e.g. scraping it off of a website or any service providing such information that can be called through an API. Using latitude and longitude to obtain zip codes was considered as demographic information like median income can be obtained from zip codes and could have been used as predictors to investigate if demographics of the location influence the application outcome. However, our research revealed that although zip codes can be obtained using Google API, the same service cannot be used to obtain demographic information. Keeping in mind the project timeline it wasn’t feasible to create two solutions, one for obtaining the zip codes and another for obtaining demographic information.

9. Conclusion

Despite the unbalanced nature of the dataset, Logistic Regression and Random Forest give a fair predictive model proving their robustness to this anomaly. Naive Bayes gives an unusable predictive model and this could be due to violation of probabilities independence assumption of Naive Bayes in this dataset or its lack of robustness to unbalanced datasets as compared to Logistic Regression. Techniques like down sampling can be tried to improve these models. In terms of scalability, as dataset size increases Logistic Regression and Random Forest scale well. As number of nodes is increased, Naive Bayes and Random Forest show improved performance. Considering both the predictive capability and scalability, Random Forest is the most suitable model for this dataset.