**PROJECT REPORT**

**ON**

**H1-B DATA SET 2017**

(Dataset Provided by US Department of Labor about H1B Visa - 2017)



**Team Number: 15**

**Members:**

Sriram Ravindran - 001003684

Ajith Babu Ramesh – 001061464

**Abstract**

The report consists of our conclusion on H1-B Predictions in the United States of America based on the Supervised Machine Learning Algorithms.

We used a real-time dataset from Kaggle. The initial process includes visualizing the raw data followed by cleaning the data based on our requirement and once again visualizing the cleaned data and finally implementing various models to examine its performance.

The research focuses on different algorithms that make use of classifiers such as

* Decision Tree Classifier
* Logistic Regression Classifier
* Gaussian Naïve Bayes Classifier
* Gradient Boost Classifier
* Random Forest Classifier
* Ada Boost Classifier
* Multinomial Naïve Bayes Classifier

Finally predicting the accuracy score of each algorithm and determining which algorithm will be suitable for our prediction.

The process also involves developing a front-end using Streamlit for the ease of user access, so that the prediction can be determined with one click.

**Table of Contents**

Contents

[Introduction 3](#_Toc69670404)

[Background: 4](#_Toc69670405)

[Motivation: 4](#_Toc69670406)

[Goal: 4](#_Toc69670407)

[Methodologies and Algorithms 4](#_Toc69670408)

[Dataset Review: 5](#_Toc69670409)

[Libraries and Software’s used: 5](#_Toc69670410)

[Data Cleaning Process: 5](#_Toc69670411)

[Feature Selection: 6](#_Toc69670412)

[Algorithms Used: 6](#_Toc69670413)

[Dataset description 7](#_Toc69670414)

[Data Source 13](#_Toc69670415)

[Result and Analysis: 14](#_Toc69670416)

[Analysis 14](#_Toc69670417)

[Data pre-processing: 18](#_Toc69670418)

[Feature Engineering: 20](#_Toc69670419)

[Data Visualization after Cleaning: 22](#_Toc69670420)

[Model Training: 24](#_Toc69670421)

[Streamlit: 27](#_Toc69670422)

[Conclusion: 28](#_Toc69670423)

[References 29](#_Toc69670424)

# Introduction

The H-1B is a visa in the United States under the Immigration and Nationality Act, section 101(a)(15)(H) that allows U.S. employers to temporarily employ foreign workers in specialty occupations.  The duration of stay is three years, extendable to six years; after which the visa holder may need to reapply. Laws limit the number of H-1B visas that are issued each year.

## Background:

This system aims to predict the eligibility of a person to go through the H-1B visa process with the dataset that was chosen. This prediction system has ‘Case status’ as the target variable and its predictions are based on a variety of features such as Employer Country, Job Title, Full/Part-time, Employee Wage.

## Motivation:

Getting an H-1B was a dream for many techies in non-US countries. We both being software developers at Cognizant Technology Solutions were confused on the pattern of eligibility criteria of people for H-1B. This motivated us to work on this project to analyse which type of employees are eligible to undergo the H-1B visa process.

## Goal:

Our goal is to predict whether a person is eligible to go through the H-1B visa process. We have the target variable as **Case Status** based on which we identify a person’s eligibility.

# Methodologies and Algorithms

The target variable is “**Case Status**” which contains 2 status meaning:

* Certified – Status associated with the last significant event or decision (Certified)
* Denied – Status associated with the last significant event or decision (Denied)

The prediction model used here is **Classification**.

## Dataset Review:

This dataset contains H-1B details for the year from 2011 to 2017, with approximately 0.6 million records overall. The dataset consists of one csv file which is around 240 MB and has total 52 features. We have selected certain important features to be given as the input to the model to predict the H-1B Status.

## Libraries and Software’s used:

* Numpy
* Pandas
* Matplotlib
* Seaborn
* Plotly Express
* Metrics
* Precision score
* Recall
* f1\_score
* train\_test\_split
* sklearn.ensemble
* sklearn.tree
* imblearn.combine
* sklearn.model\_selection
* SMOTEENN
* TargetEncoder
* Pipeline
* Streamlit
* Spyder
* collections

## Data Cleaning Process:

The Raw Dataset had many Nan, irrelevant values that can impact the model accuracy when injected. Hence, we found certain appropriate way to get rid of these values.

## Feature Selection:

The dataset had 52 Columns that can clatter the model, since it had many irrelevant and empty values in it. Though there were more columns in this dataset, many features clearly did not have any meaning in including in our prediction model. Hence, to get an accurate model, we built our final model that had totally 10 columns in it. We have explained in detail in the later sections as why we have selected only 10 features out of 52 features.

## Algorithms Used:

Below are the Algorithms that we implemented:

* Decision Tree Classification
* Logistic Regression Classification
* Gradient Boost Classification
* Gaussian Naïve Bayes Classification
* Ada Boost Classification
* Multinomial Naïve Bayes Classification
* Random Forest Classification

**Decision Tree Classification:**

This classifier uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. This algorithm is highly useful where there are a greater number of features. However, in our case, the model's accuracy in predicting target values was not up to par.

**Logistic Regression Classification:**

The logistic sigmoid function is used to transform the output of this classifier into a probability value. This model gave a below average accuracy of target value prediction in our case.

**Gradient Boost Classification:**

Gradient boosting is a [machine learning](https://en.wikipedia.org/wiki/Machine_learning) technique for regression and classification problems, which produces a prediction model in the form of an [ensemble](https://en.wikipedia.org/wiki/Ensemble_learning) of weak prediction models, typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient boosted trees, which usually outperforms random forest. In our case, this classification gave the highest ROC\_AUC Score.

**Gaussian Naïve Bayes Classification:**

They are among the simplest Bayesian network models but coupled with kernel density estimation, they can achieve higher accuracy levels. This classifier is based on the Naïve Bayes Theorem. When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a [normal](https://en.wikipedia.org/wiki/Normal_distribution) (or Gaussian) distribution.

**AdaBoost Classification:**

 Boosting methods train the predictors sequentially, each trying to correct its predecessor. At a high level, AdaBoost is like Random Forest in that they both tally up the predictions made by each decision trees within the forest to decide on the final classification. In our prediction, this classifier almost gave the best ROC\_AUC score, but was less than Gradient Boost.

**Random Forest Classification:**

This classifier operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) of the individual trees. In our prediction Random forest gave a decent ROC\_AUC score.

**Multinomial Naïve Bayes Classification:**

With a multinomial event model, samples (feature vectors) represent the frequencies with which certain events have been generated by a [multinomial](https://en.wikipedia.org/wiki/Multinomial_distribution) is the probability that event i occurs (or K such multinomials in the multiclass case). In our case this classifier gave a very average prediction.

# Dataset description

This dataset contains H-1B details from 2011 to 2017, with approximately 0.6 million records overall. The dataset consists of one csv file which is around 240 MB and has total 52 features, of which:

* 26 attributes are String,
* 11 attributes are Integer,
* 6 attributes are Boolean,
* And 9 Other.

The columns are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Features** | **Description** | **Null values** |
|  | CASE\_NUMBER | Unique identifier assigned to each application submitted for processing  to the Chicago National Processing Center. | No |
|  | CASE\_STATUS | Status associated with  the last significant event or decision. Valid values include “Certified”, “Denied” | No |
|  | CASE\_SUBMITTED | Date and time the application was submitted. | No |
|  | DECISION\_DATE | Date on which the last significant event or decision was recorded. | No |
|  | VISA\_CLASS | Indicates the type of temporary application submitted for processing. R =  H-1B; A = E-3 Australian; C = H-1B1 Chile; S = H-1B1 Singapore. Also  referred to as “Program” in prior years. | No |
|  | EMPLOYMENT\_START\_DATE | Beginning date of employment. | Yes |
|  | EMPLOYMENT\_END\_DATE | Ending date of  employment. | Yes |
|  | EMPLOYER\_NAME | Name of employer submitting labor condition application. | Yes |
|  | EMPLOYER\_BUSINESS\_DBA | Trade Name or dba name of employer submitting labor condition  application, if applicable. | Yes |
|  | EMPLOYER\_ADDRESS | Contact information of the Employer requesting temporary labor  certification. | Yes |
|  | EMPLOYER\_CITY | Yes |
|  | EMPLOYER\_STATE | Yes |
|  | EMPLOYER\_POSTAL\_CODE | Yes |
|  | EMPLOYER\_COUNTRY | Yes |
|  | EMPLOYER\_PROVINCE | Yes |
|  | EMPLOYER\_PHONE | Yes |
|  | EMPLOYER\_PHONE\_EXT | Yes |
|  | AGENT\_REPRESENTING\_EMPLOYER | Y = Employer is represented by an Agent or Attorney; N = Employer is  not represented by an Agent or Attorney. | Yes |
|  | AGENT\_ATTORNEY\_NAME | Name of Agent or Attorney filing an H-1B application on behalf of the  employer. | No |
|  | AGENT\_ATTORNEY\_CITY | City information for the Agent or Attorney filing an H-1B application on  behalf of the employer. | Yes |
|  | AGENT\_ATTORNEY\_STATE | State information for the Agent or Attorney filing an H-1B application on  behalf of the employer. | Yes |
|  | JOB\_TITLE | Title of the job. | Yes |
|  | SOC\_CODE | Occupational code associated with the job being requested for  temporary labor condition, as classified by the Standard Occupational  Classification (SOC) System. | Yes |
|  | SOC\_NAME | Occupational name associated with the SOC\_CODE. | Yes |
|  | NAICS\_CODE | Industry code associated with the employer requesting permanent labor  condition, as classified by the North American Industrial Classification  System (NAICS). | Yes |
|  | TOTAL\_WORKERS | Total number of foreign workers requested by the Employer(s). | No |
|  | NEW\_EMPLOYMENT | Indicates requested worker(s) will begin employment for new employer,  as defined by USCIS I-29. | No |
|  | CONTINUED\_EMPLOYMENT | Indicates requested worker(s) will be continuing employment with same  employer, as defined by USCIS I-29. | No |
|  | 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. | No |
|  | NEW\_CONCURRENT\_EMPLOYMENT | Indicates requested worker(s) will begin employment with additional  employer, as defined by USCIS I-29. | No |
|  | CHANGE\_EMPLOYER | Indicates requested worker(s) will begin employment for new employer,  using the same classification currently held, as defined by USCIS I-29. | No |
|  | 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. | No |
|  | FULL\_TIME\_POSITION | Y = Full Time Position; N = Part Time Position. | Yes |
|  | PREVAILING\_WAGE | Prevailing Wage for the job being requested for temporary labor  condition. | Yes |
|  | PW\_UNIT\_OF\_PAY | Unit of Pay. Valid values include “Daily (DAI),” “Hourly (HR),” “Bi-weekly  (BI),” “Weekly (WK),” “Monthly (MTH),” and “Yearly (YR)”. | Yes |
|  | PW\_WAGE\_LEVEL | Variables include "I", "II", "III", "IV" or "N/A." | Yes |
|  | PW\_SOURCE | Variables include "OES", "CBA", "DBA", "SCA" or "Other". | Yes |
|  | PW\_SOURCE\_YEAR | Year the Prevailing Wage Source was Issued. | Yes |
|  | PW\_SOURCE\_OTHER | If "Other Wage Source", provide the source of wage. | Yes |
|  | WAGE\_RATE\_OF\_PAY\_FROM | Employer’s proposed wage rate. | No |
|  | WAGE\_RATE\_OF\_PAY\_TO | Maximum proposed wage rate. | Yes |
|  | WAGE\_UNIT\_OF\_PAY | Unit of pay. Valid values include “Hour", "Week", "Bi-Weekly", "Month",  or "Year". | Yes |
|  | H-1B\_DEPENDENT | Y = Employer is H-1B Dependent; N = Employer is not H-1B Dependent. | Yes |
|  | WILLFUL\_VIOLATOR | Y = Employer has been previously found to be a Willful Violator; N =  Employer has not been considered a Willful Violator. | Yes |
|  | 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); | Yes |
|  | 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. | Yes |
|  | PUBLIC\_DISCLOSURE\_LOCATION | Variables include "Place of Business" or "Place of Employment." | Yes |
|  | WORKSITE\_CITY | City information of the foreign worker's intended area of employment. | Yes |
|  | WORKSITE\_COUNTY | County information of the foreign worker's intended area of employment. | Yes |
|  | WORKSITE\_STATE | State information of the foreign worker's intended area of employment. | Yes |
|  | WORKSITE\_POSTAL\_CODE | Zip Code information of the foreign worker's intended area of  employment. | Yes |
|  | ORIGINAL\_CERT\_DATE | Original Certification Date for a Certified\_Withdrawn application. | Yes |

## Data Source

The dataset is taken from Kaggle:

**Name:** H-1B Data Set 2017

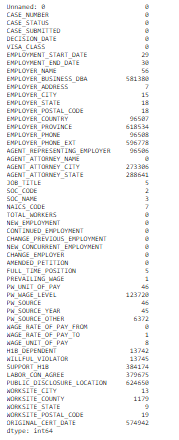
**Link:** <https://www.kaggle.com/jonamjar/h1b-data-set-2017>

# Result and Analysis:

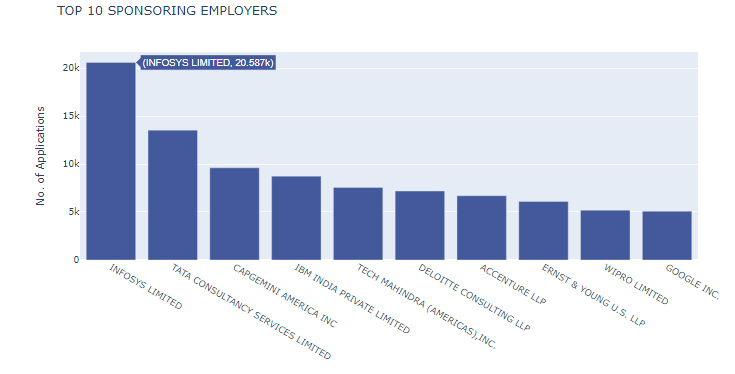
## Analysis

Since the Data is huge, we tried to analyse initially with the raw dataset, which gave us an idea of how the data is and what futures we can consider can predict the H-1B approval status.

Initially, we found how many features had null values in them,



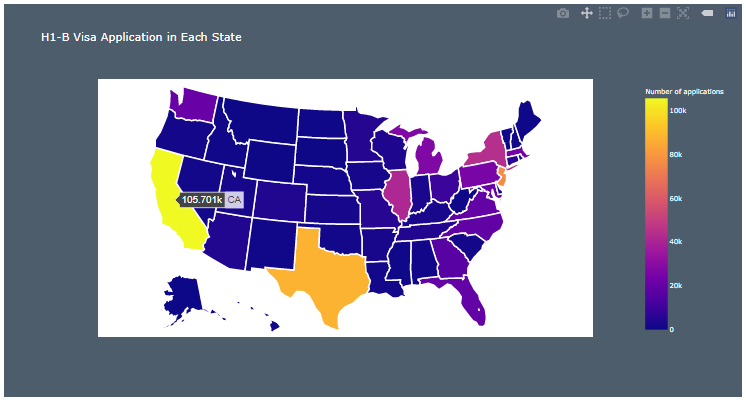
Then we did few Visualizations with the raw data,



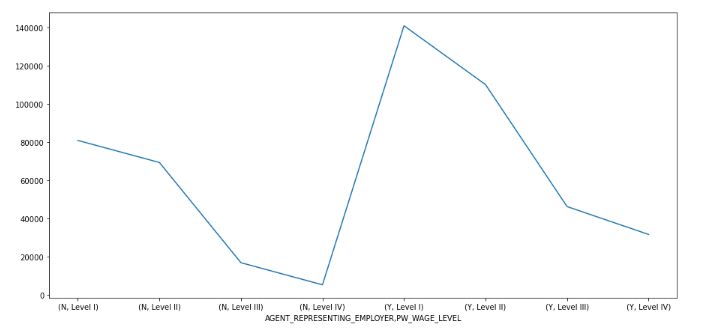
The above bar chart tells us which Employer is sponsoring a greater number of H-1B applications. Since we had many Employers in our Dataset, we reduced it to Top-10 Employers. We could see that Infosys Limited is sponsoring about 20.5k Applications.



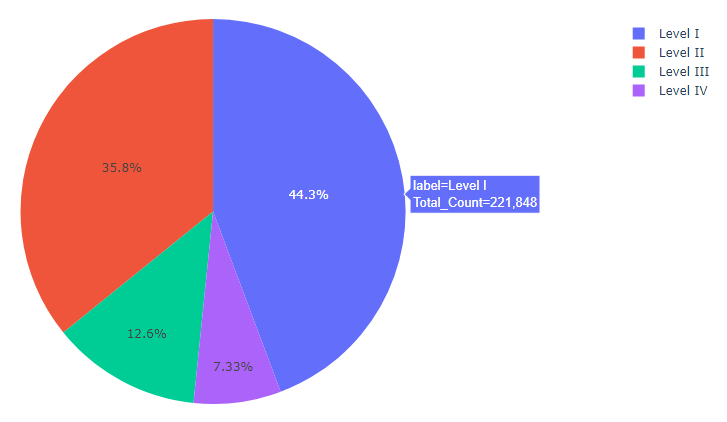
Here we grouped the Top Employers with their Case Status, here we could again see that Infosys tops the table with around 20.5k Certified Cases.

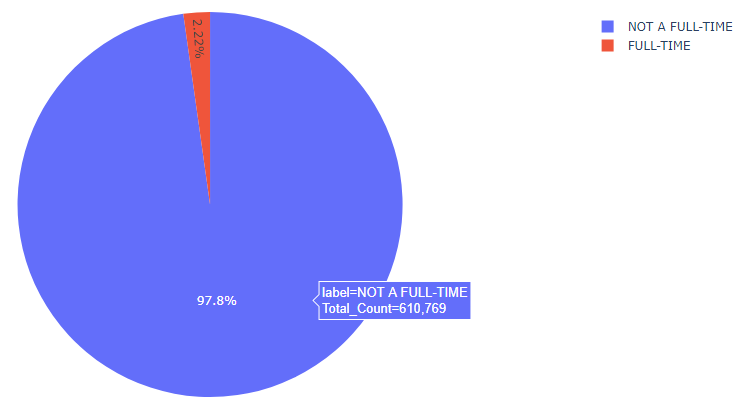


From the above graph, we could see that California has the most H1-B Visa Applications of around 105.7K.



The Above graph shows the prevailing wages grouped by the Employee’s wage level and Agent representing the Employer value.





## Data pre-processing:

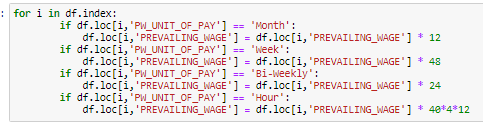
A picture containing website

Description automatically generated

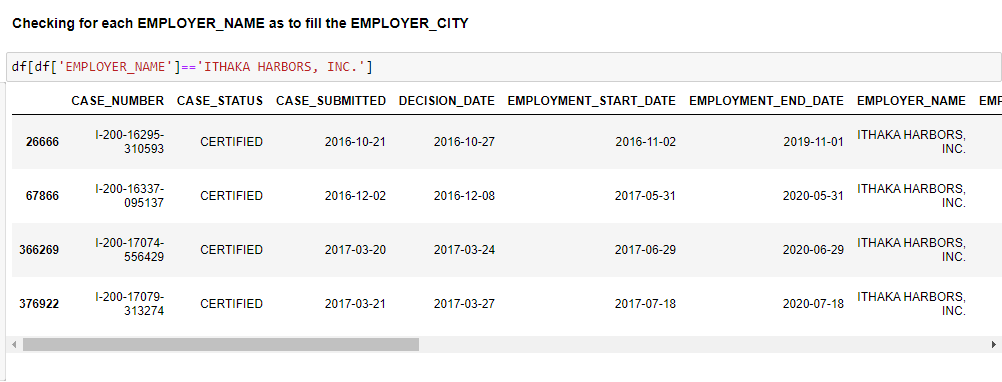
We considered only the Visa class as **H-1B** and the country as **United States of America**.

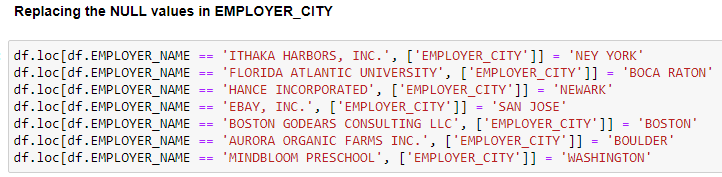


Then upon analysing the data we saw that the Pay period of the employees was spread as Hour, Week, Month, Bi-Weekly, and yearly. Hence, we made everything to be Year by calculating the values as follows:



Then we removed most of the null values by replacing them with the values that are already present in the dataset or by taking the mode of each column or by removing the record if the records are very small in number compared to the actual dataset.





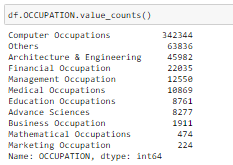
Upon checking the Correlation matrix, we considered only those features that will contribute to prediction. E.g.: we considered only the country and state and discarded phone numbers, zip codes, etc from Employer’s perspective.

We also combined the values of Job Description from Job Title, SOC name, and SOC Code and created a new group from all the above and saved it as Occupation.

## Feature Engineering:

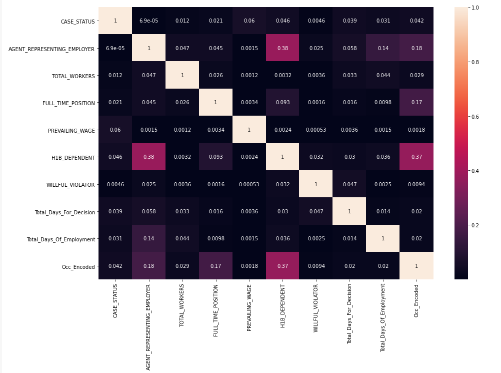
* We created **Total\_Days\_For\_Decision** from case status submitted date and decision.
* Created **Total\_Days\_Of\_Employment** from employment start date and end date
* Based on SOC Name we categorised and created a new feature as Occupation,





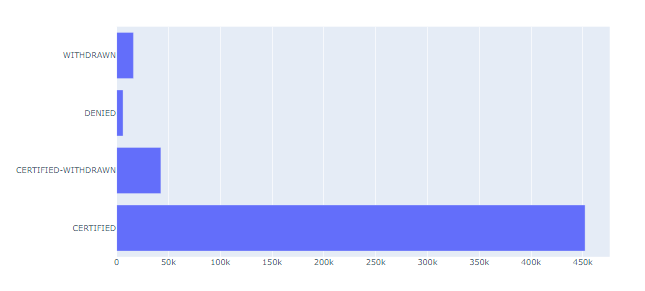
Finally, after removing all the unwanted values, and considering only those features that are needed for prediction the features and correlation matrix are as follows,

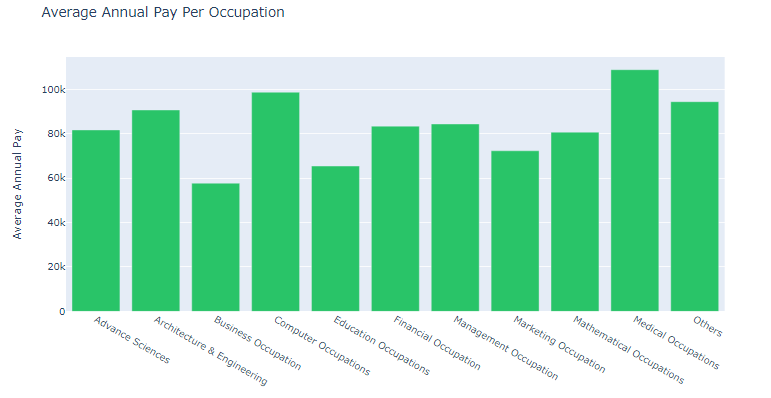


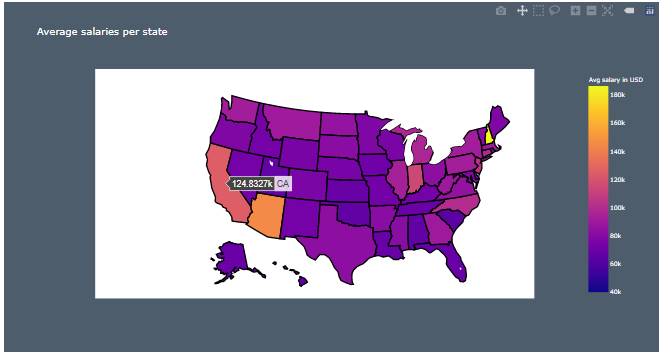


## Data Visualization after Cleaning:

Finally, after Cleaning the Data, we visualised few Stats,

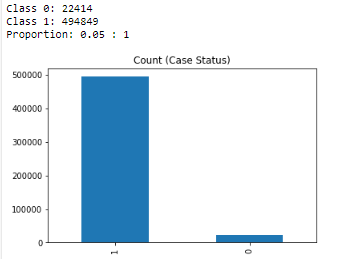






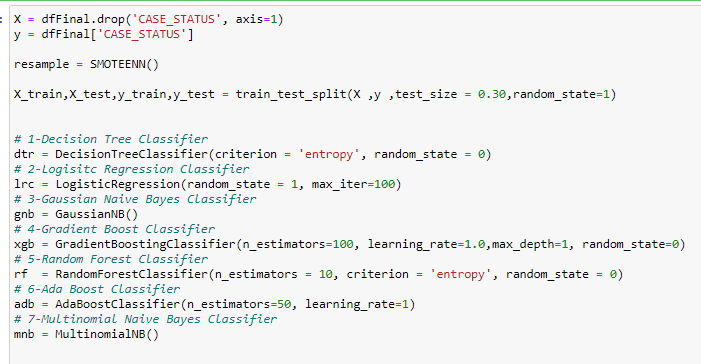
We could see the California again tops with average salary around 124.8K

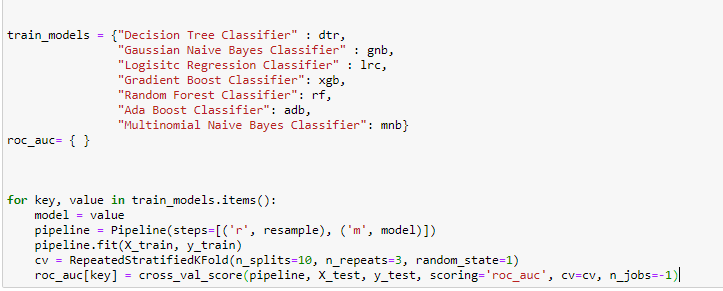
We also visualized that the target variable for this dataset is highly imbalanced,



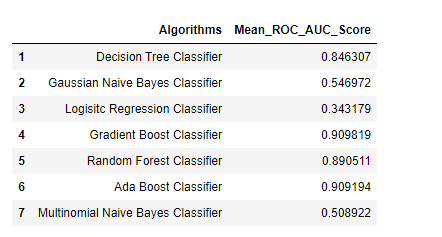
## Model Training:

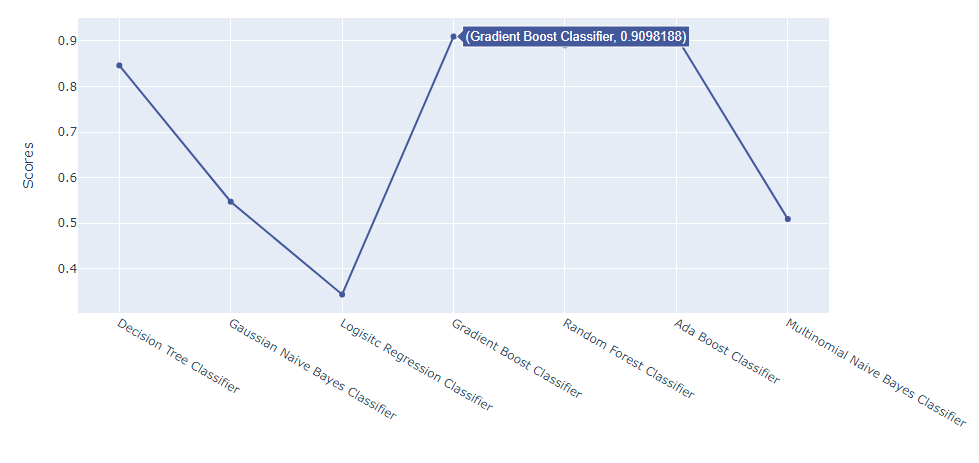
Since the target variable was highly imbalanced, we resampled using SMOTEENN and then injected those values along with model in a pipeline. We split the test-size and train-size as 30 and 70 respectively,



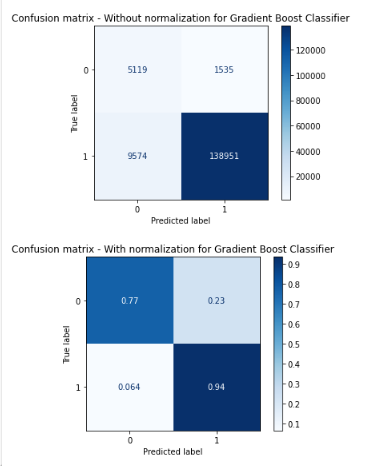


Below is the ROC\_AUC scores calculated for Each of the Algorithms used,



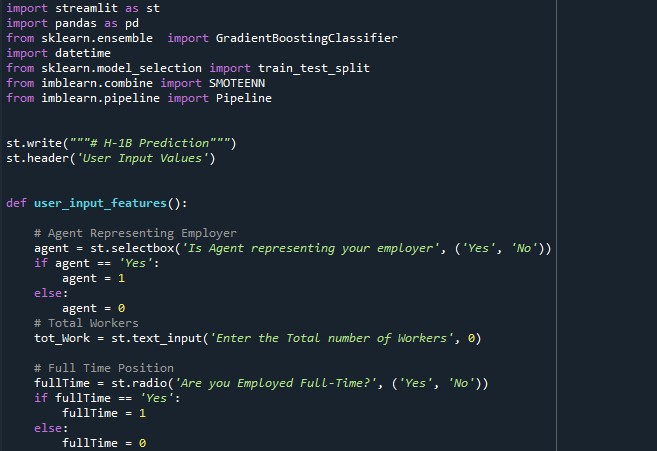


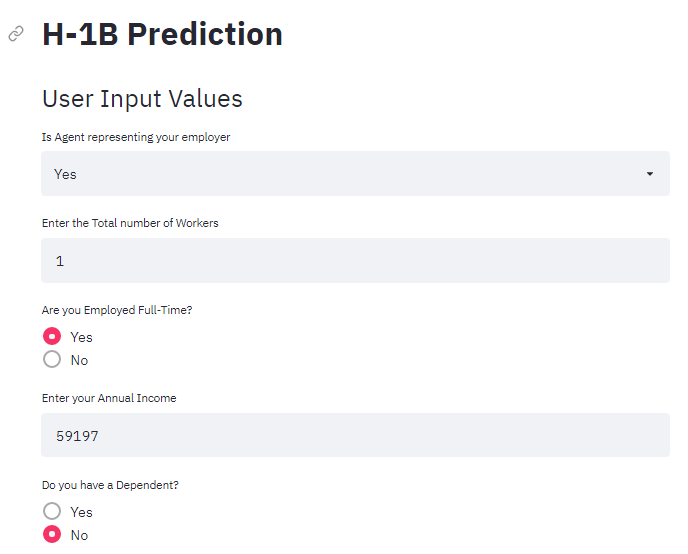
Finally, we plotted the Confusion Matrix for the Gradient Boost Classifier with and without Normalization,

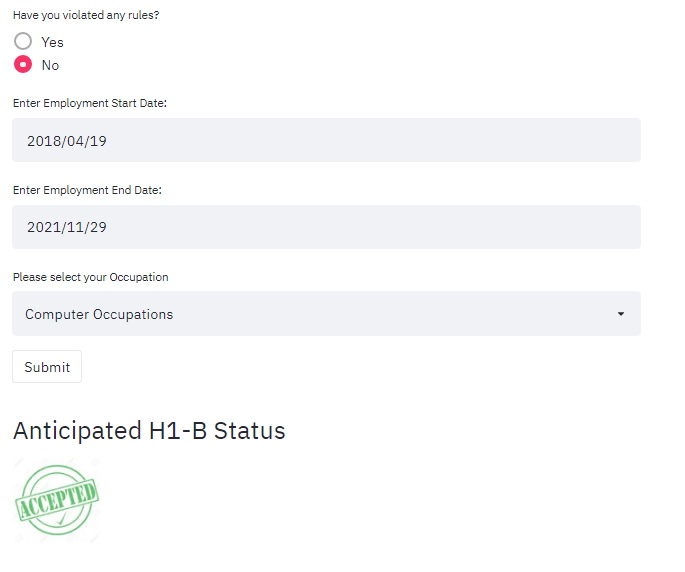


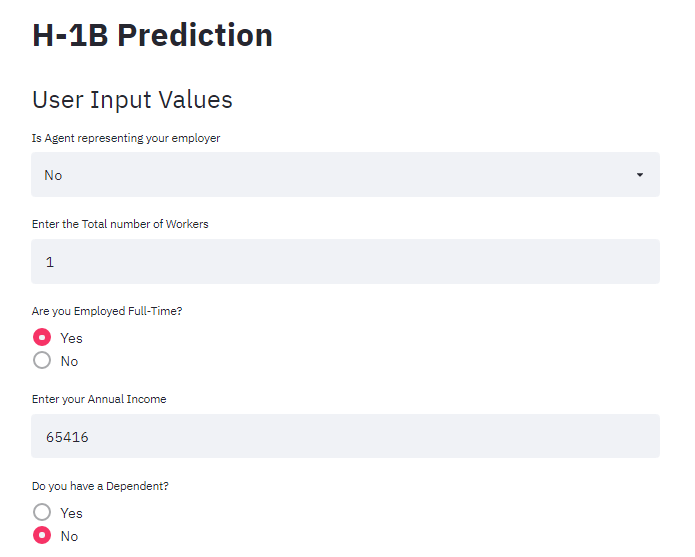
## Streamlit:

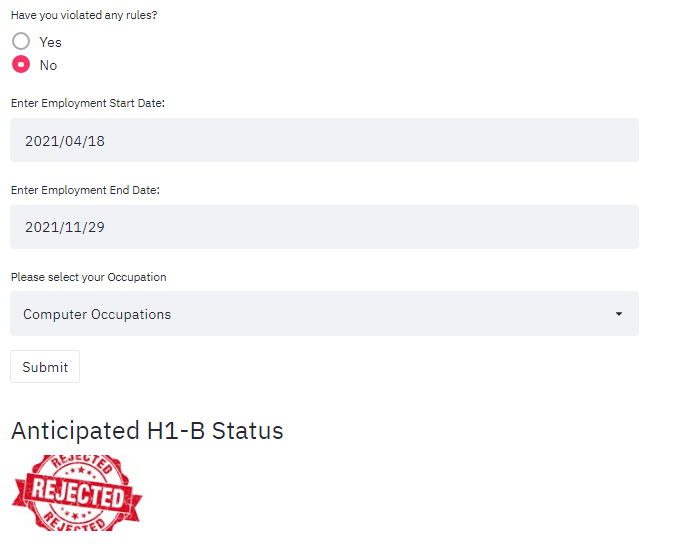
We also developed a streamlit application for the final Classifier Method which can predict the H-1B Approval/ Denial status.











# Conclusion:

From the ROC\_AUC scores of all the algorithms, the Gradient Boost Classifier produced a significant score of accuracy for prediction of the Target Variable (Case Status).

# References

<https://www.kaggle.com/jonamjar/h1b-data-set-2017?select=H-1B_Disclosure_Data_FY17.csv>

<https://stackoverflow.com/>

<https://machinelearningmastery.com/>

<https://github.com/>

<https://scikit-learn.org/>

<https://plotly.com/>

<https://streamlit.io/>

<https://en.wikipedia.org/wiki/Main_Page>