2020

Capstone project

Group 2

“Predicting the Outcome of H-1B Visa Applications”,

Great Lakes Institute Of Management, Bilaspur Tauru Road, Near Bilaspur Chowk, Gurgaon - 122413, Haryana, India, Phone: +91-124-2865800

Title of capstone project

Predicting the Outcome of H-1B Visa Applications

Submitted towards partial fulfilment of the criteria

For award of PGPDSE

By GLIM

Prepared & Submitted by:

1. **Aarti Gupta**
2. **Shardul Jadhav**
3. **Saad Khan**
4. **Sumit Gaikwad**
5. **Shreekrishna Jadhav**
6. **Mahendra Jogdankar**

Mentored and Guided by:

Srikar Muppidi

Batch: August 2019

- **Techniques**: Predictive Modelling and Data Mining
- **Tools**: Python and Tableau
- **Domain**: US H-1B Visa

Great Lakes Institute Of Management, Bilaspur Tauru Road, Near Bilaspur Chowk, Gurgaon - 122413, Haryana, India
Phone: +91-124-2865800

ACKNOWLEDGEMENT

We hereby certify that the work done by us for the implementation and completion of this project is absolutely original and to the best of our knowledge. It is a team effort and each of the member has equally contributed in the project.

Date: 22nd January 2020

Place: Mumbai

CERTIFICATE OF COMPLETION

This is to certify that the project titled "Predicting the Outcome of H-1B Visa Applications", for case resolution was undertaken and completed under the supervision of Mr. Srikar Muppidi for Post Graduate Program in Data Science and Engineering (PGP – DSE)

Mentor: Mr. Srikar Muppidi

Contents

I. Introduction

1. Brief Introduction

In our project, we aim to predict the outcome of H-1B visa applications that are filed by many high-skilled foreign nationals every year. We framed the problem as a classification problem and applied supervised classification models like Naive Bayes, Logistic Regression, KNN and ensemble technique like Random Forest, Decision Tree, in order to output a predicted case status of the application. The input to our algorithm is the attributes of the applicant which will be further explained in the following parts.

2. Structure

H-1B is a type of non-immigrant visa in the United States that allows foreign nationals to work in occupations that require specialized knowledge and a bachelor's degree or higher in the specific specialty [1]. This visa requires the applicant to have a job offer from an employer in the US before they can file an application to the US immigration service (USCIS). USCIS grants 85,000 H-1B visas every year, even though the number of applicants far exceed that number [2]. The selection process is claimed to be based on a lottery, hence how the attributes of the applicants affect the final outcome is unclear. We believe that this prediction algorithm could be a useful resource both for the future H-1B visa applicants and the employers who are considering to sponsor them.

3. Step involved in H-1B visa application selection **Figure 1 [3]**:

- The first step of the H-1B application process is for the U.S. employer to file the H-1B petition on behalf of the foreign worker.
- In the second step, the prevailing and actual wages should be confirmed by the State Employment Security Agency. If the prevailing wage exceeds the offer made by the prospective employer then a wage determination will be sought.
- The third step of the H-1B application process is to file the Labour Condition Application. The next step is to prepare the petition and file it at the proper USCIS office.
- Processing times for H-1B application petitions are subject to vary from location to location. If you would like your petition expedited you may elect for premium processing.
- The final step of the H-1B application process is to check the status of your H-1B visa petition by entering your receipt number. Once USCIS has your application on file, they will update your status on their system.

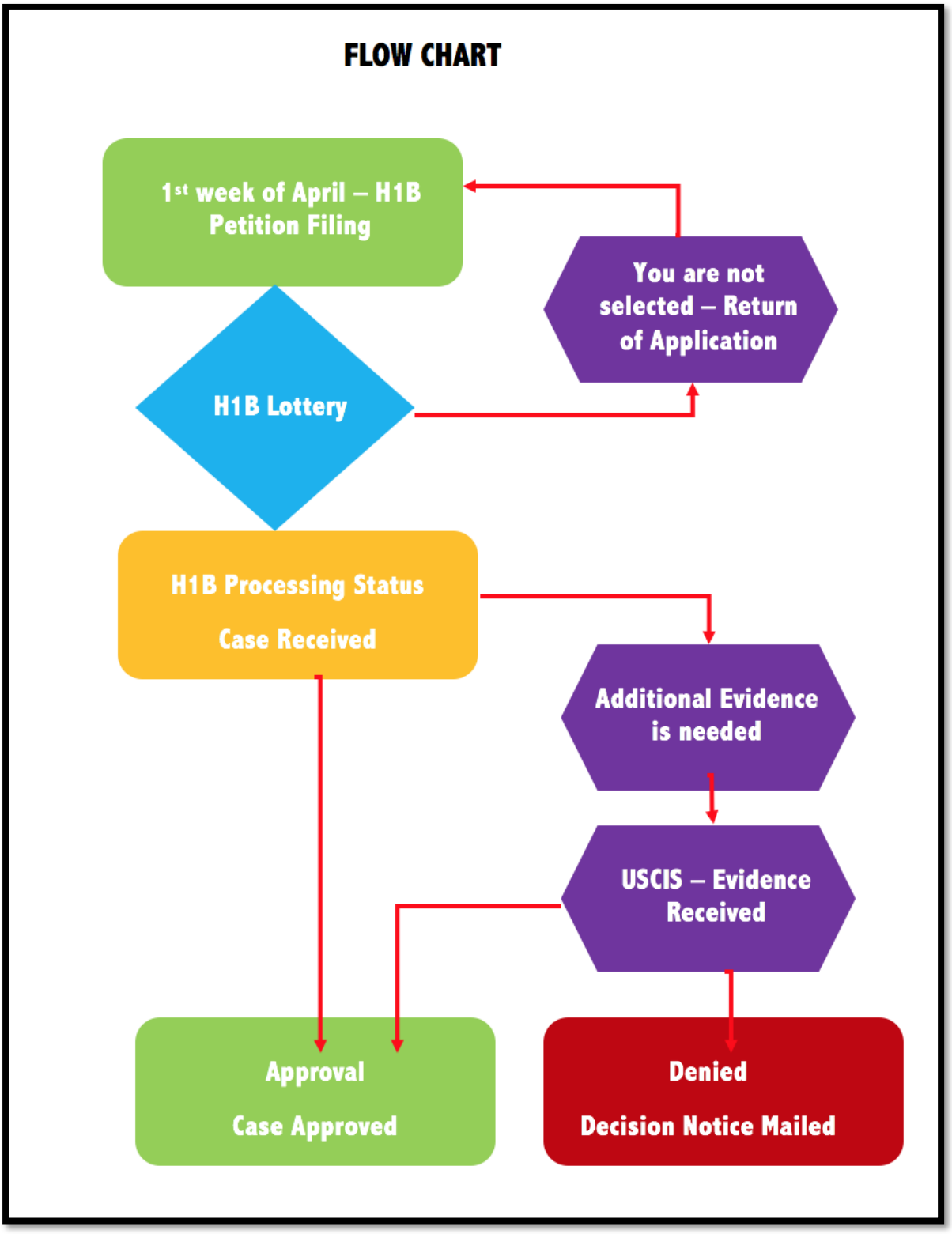

Figure 1: Summary of steps involved in H-1B application selection

II. Dataset, Feature Selection and Transformation

Our dataset is from '**data.world**' listed under the name "**H-1B_Disclosure_Data_FY17**" [4], and it initially included about 5.2 million data points. It contained 39 features and 1 label which can be examined in **Figure 2**.

CASE_NUMBER	CASE_STATUS	CASE_SUBMITTED	DECISION_DATE	VISA_CLASS	EMPLOYMENT_START_DATE	EMPLOYMENT_END_DATE
I-200-16055-173457	CERTIFIED-WITHDRAWN	2/24/2016	10/1/2016	H-1B	8/10/2016	8/10/2019
I-200-16064-557834	CERTIFIED-WITHDRAWN	3/4/2016	10/1/2016	H-1B	8/16/2016	8/16/2019

EMPLOYER_NAME	EMPLOYER_ADDRESS	EMPLOYER_CITY	EMPLOYER_STATE	EMPLOYER_POSTAL_CODE	EMPLOYER_COUNTRY	EMPLOYER_PROVINCE
DISCOVER PRODUCTS INC.	2500 LAKE COOK ROAD	RIVERWOODS	IL	60015	UNITED STATES OF AMERICA	NaN
DFS SERVICES LLC	2500 LAKE COOK ROAD	RIVERWOODS	IL	60015	UNITED STATES OF AMERICA	NaN

EMPLOYER_PHONE	EMPLOYER_PHONE_EXT	AGENT_ATTORNEY_NAME	AGENT_ATTORNEY_CITY	AGENT_ATTORNEY_STATE	JOB_TITLE	SOC_CODE
2.24405e+09	NaN	ELLSWORTH, CHAD	NEW YORK	NY	ASSOCIATE DATA INTEGRATION	15-1121
2.24405e+09	NaN	ELLSWORTH, CHAD	NEW YORK	NY	SENIOR ASSOCIATE	15-2031

	SOC_NAME	NAICS_CODE	TOTAL_WORKERS	FULL_TIME_POSITION	PREVAILING_WAGE	PW_UNIT_OF_PAY	PW_SOURCE
COMPUTER SYSTEMS ANALYSTS		522210.0	1	Y	59,197.00	Year	OES
OPERATIONS RESEARCH ANALYSTS		522210.0	1	Y	49,800.00	Year	Other

PW_SOURCE_YEAR	PW_SOURCE_OTHER	WAGE_RATE_OF_PAY_FROM	WAGE_RATE_OF_PAY_TO	WAGE_UNIT_OF_PAY	H-1B_DEPENDENT	WILLFUL_VIOLATOR
2015.0	OFLC ONLINE DATA CENTER	65,811.00	67,320.00	Year	N	N
2015.0	TOWERS WATSON DATA SERVICES 2015 CSR PROFESSIO...	53,000.00	57,200.00	Year	N	N

WORKSITE_CITY	WORKSITE_COUNTY	WORKSITE_STATE	WORKSITE_POSTAL_CODE	ORIGINAL_CERT_DATE
RIVERWOODS	LAKE	IL	60015	3/1/2016
RIVERWOODS	LAKE	IL	60015	3/8/2016

Figure 2: Two data points from the unprocessed dataset

1. Pre-processing of Dataset

- We processed some of the existing features, created new features that we thought could be useful for prediction and discarded some features using the library Pandas.

A. We have discarded features which hold more than 90 % of null values [5]**.**Therefore dropping features '**EMPLOYER_PROVINCE**', '**EMPLOYER_PHONE_EXT**', and '**ORIGINAL_CERT_DATE**' from dataset. **Figure 3**

EMPLOYER_PROVINCE	99.07
EMPLOYER_PHONE_EXT	95.75
ORIGINAL_CERT_DATE	93.15

Figure 3: Features more than 90% null values

B. Also dropping features which was having unique values and other features which not significant for predicting model, features like '**CASE_NUMBER'**, **'EMPLOYER_POSTAL_CODE', 'EMPLOYER_PHONE', 'EMPLOYER_PHONE_EXT', 'NAICS_CODE', AGENT_ATTORNEY_CITY', 'AGENT_ATTORNEY_STATE', 'PW_SOURCE', 'PW_SOURCE_OTHER ', 'WORKSITE_CITY', 'WORKSITE_COUNTY', 'WORKSITE_STATE', 'WORKSITE_POSTAL_CODE**' at starting only.

2. Null Value and Anomalies Treatment

- Features which were selected for further analysis hold less than 2 % of null values. **Figure 4.**

WILLFUL_VIOLATOR	1.95
H-1B_DEPENDENT	1.95
PW_SOURCE_OTHER	0.97
WORKSITE_COUNTY	0.19
EMPLOYER_NAME	0.01
PW_UNIT_OF_PAY	0.01
PW_SOURCE	0.01
PW_SOURCE_YEAR	0.01

Figure 4: Selected Features less than 2% null values

A. Feature '**EMPLOYER_NAME**' null values was inputted with '**EMPLOYER_ADDRESS**' holding unique employer name.

B. Prevailing Wage is define as the average wage paid to similarly employed workers in the requested occupation in the area of intended employment [6]. It was determined in dataset that feature '**PREVAILING_WAGE**' has many zero values. Therefore treated zero values with median value of '**PREVALING_WAGE**' for the respective **'EMPLOYER_NAME**' and '**JOB_TITLE'** for single observation of '**EMPLOYER_NAME'**.

C. After treating '**PREVAILING_WAGE'** its value was used for imputing null values for feature '**PW_UNIT_OF_PAY'** based on feature '**PREVALING_WAGE' minimum and maximum value.**

D. Feature **'H-1B_DEPENDENT'** and **'WILLFUL_VIOLATOR'** was treated with maximum outcome values respectively for null values.

3. Feature selection and transformation

A. New Feature '**CS_DD_Duration**' were created by using feature '**DECISION_DATE**', **Figure 4** and '**CASE_SUBMITTED**' ,**Figure 5**,which tell how many days it took for processing H-1B visa.

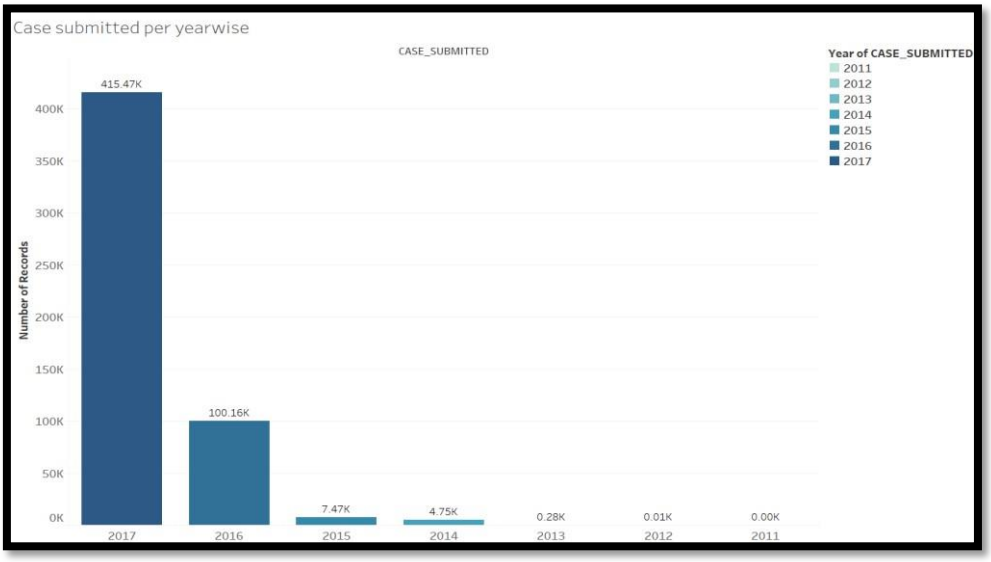

Figure 4: Case Submitted

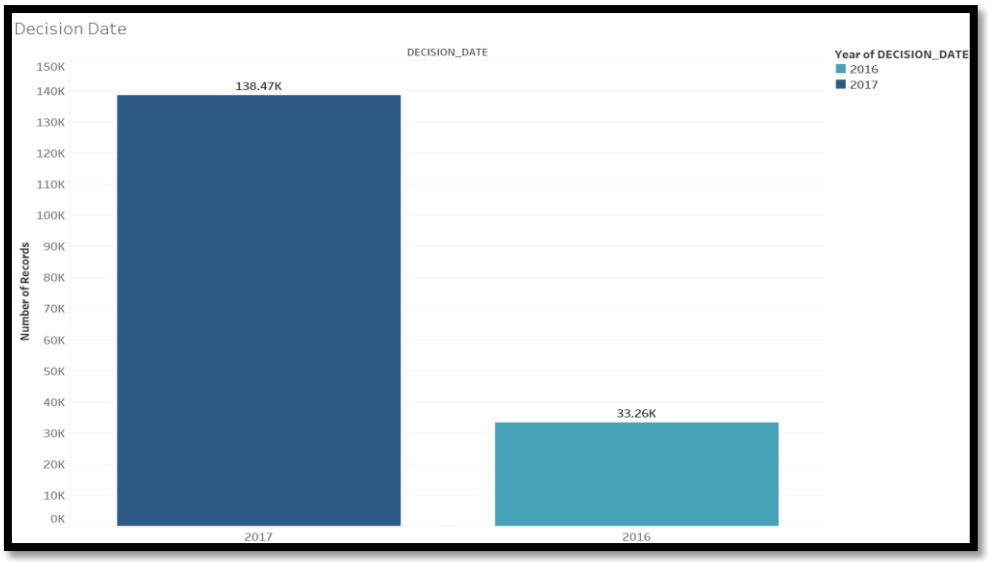

Figure 5: Decision Date

- It quite evident from **Figure 4** and **Figure 5** maximum application are filed for 2017 year.
- Having Case submitted to Decision date has 33.33% processing rate for 2017 year.

B. '**Emp_Stay_Duration_Yr**' was derive from difference between features '**EMPLOYMENT_END_DATE**', **Figure 6**,'**EMPLOYMENT_START_DATE'**, **Figure 7**, defined duration of stay of employee.

- **Figure 4** and **Figure 5**: employment duration from 2014 to 2020.

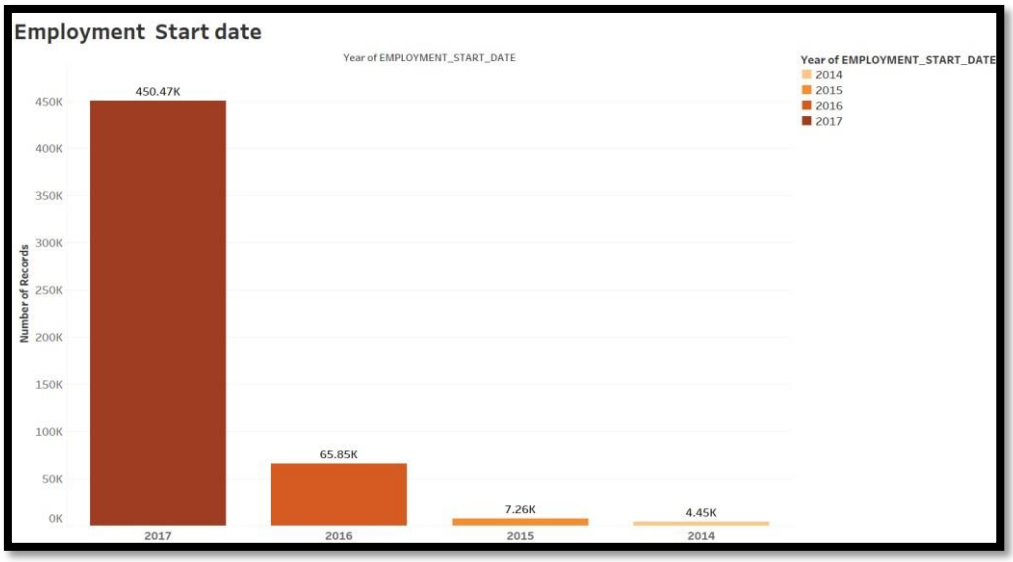

Figure 7: Employment Start date

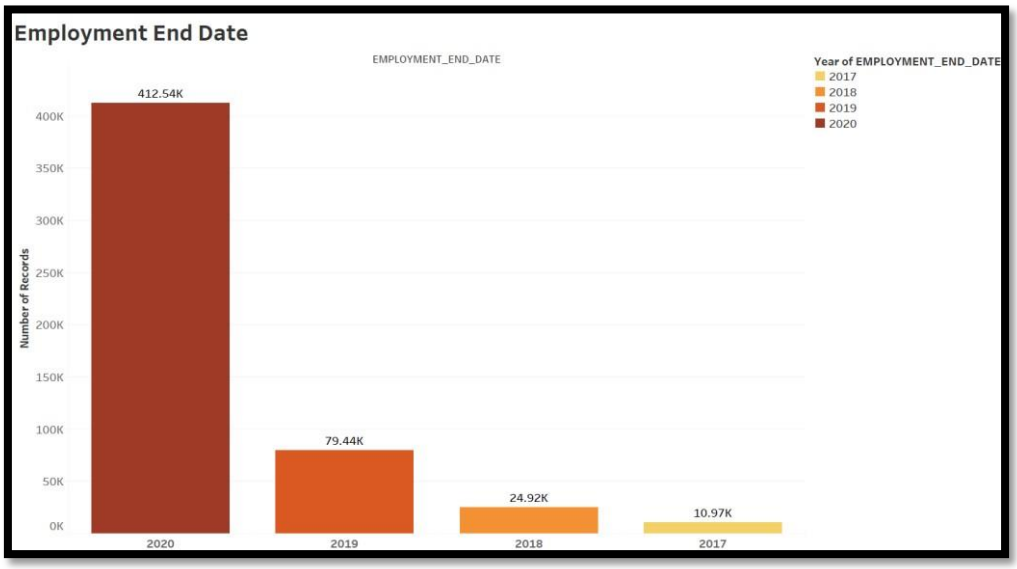

Figure 8: Employment End Date

- The initial H-1B validity period is three years from your employment start date and can be extended for another three years for 6 years [7]. **Figure 7** and **Figure 8** can be interpreted as applicant filing cases in year 2017 had validity up to year 2020. Therefore considering difference of year 2016 and 2017.

C. In particular, we noticed that the 'JOB_TITLE', feature **Figure 9** represents highly redundant information with the 'SOC_NAME' feature **Figure 10**, therefore we discarded 'JOB_TITLE' **Figure 9**.

- Also, we transformed both 'SOC_NAME' **Figure 10** and 'JOB_TITLE' features **Figure 9** into the corresponding forms of success rate and total number of applications into new feature 'OCCUPATION'.

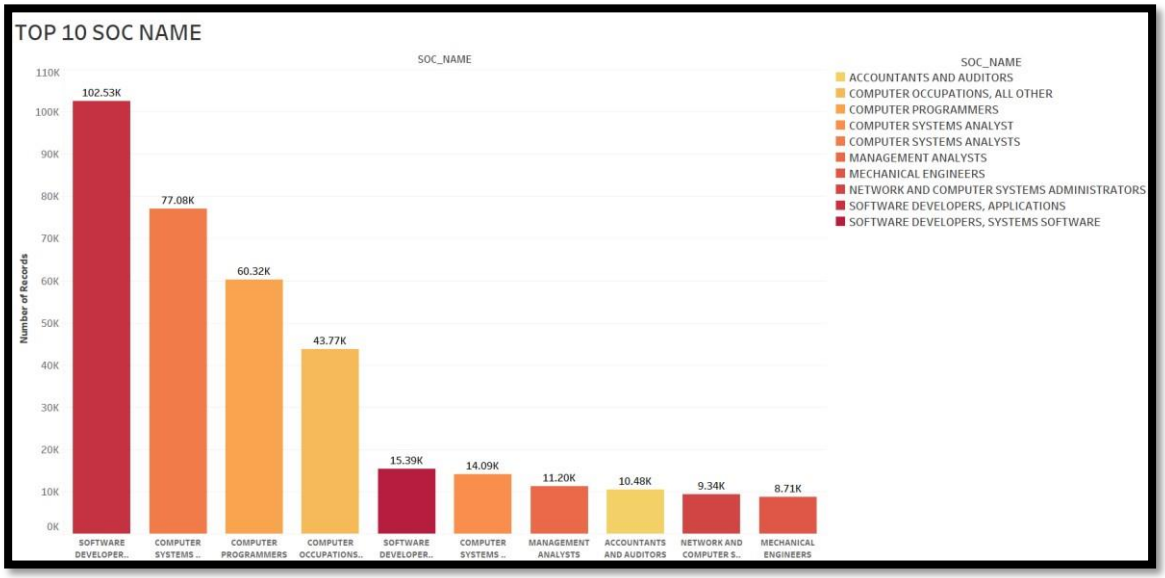

Figure 9: SOC Name

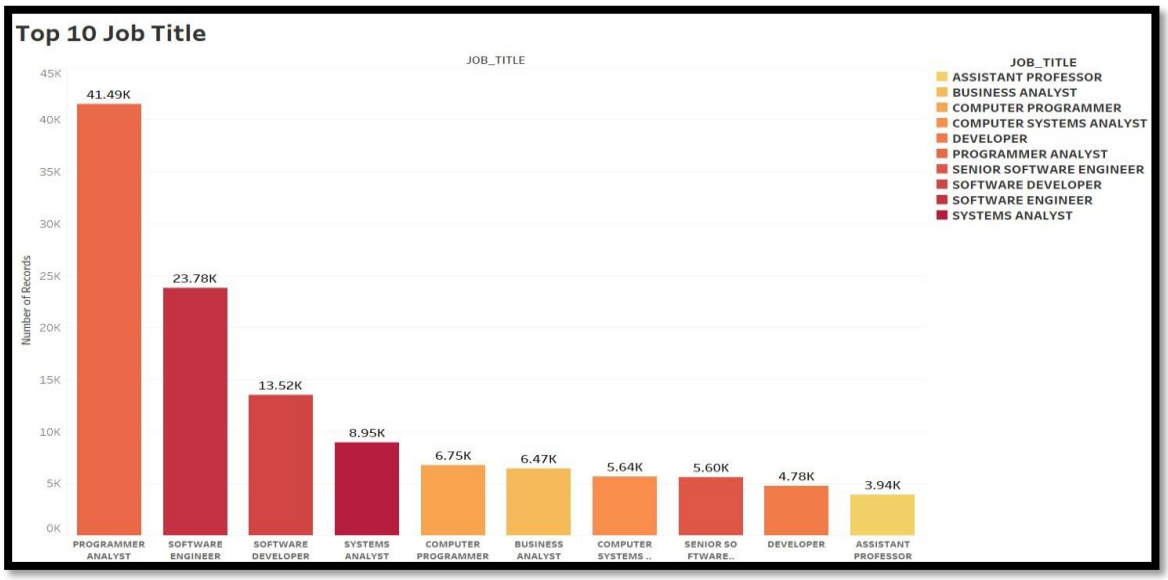

Figure 10: Job Title Name

- **Figure 9** and **Figure 10** shows that Job tile has highly redundant information with the '**SOC_NAME**' feature and can be categorize into new feature '**OCCUPTION'**.

D. '**PWGrWGFm**' feature was newly created as wage difference between prevailing wage and wage proposed by employer.

E. **''PW_UNIT_OF_PAY'** was transformed into value counts of year and hour only by converting month, week and bi-weekly with help of feature '**PREVAILING_WAGE'**.

- **PW_UNIT_OF_PAY'** consisted of value count as shown in **Figure 11**.

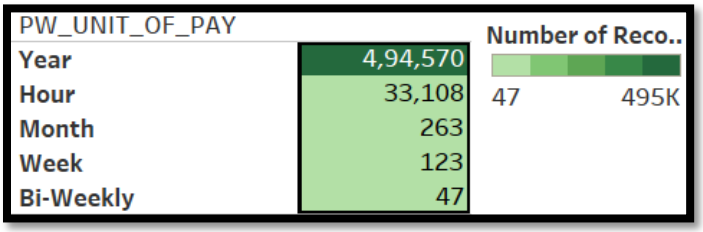

Figure 11: PW UNIT OF PAY

F. '**EMPLOYER_FREQUENCY'** by grouping by counts of '**EMPLOYER_NAME'**.

G. '**AGENT_ATTORNEY'** by using '**AGENT_ATTORNEY_NAME'** replacing value having no agent attorney name by 0 and present with name by 1.

4. Final List of Features Details with EDA:

A. **'CASE_STATUS'**:

- We excluded the cases 'CERTIFIED-WITHDRAWN' and 'WITHDRAWN', since 'WITHDRAWN' decisions are either made by the petitioning employer or the applicant, therefore not predictive of USCIS's future behaviour. **Figure 12** We labelled 'CERTIFIED' cases as 1 and 'DENIED' cases as 0.

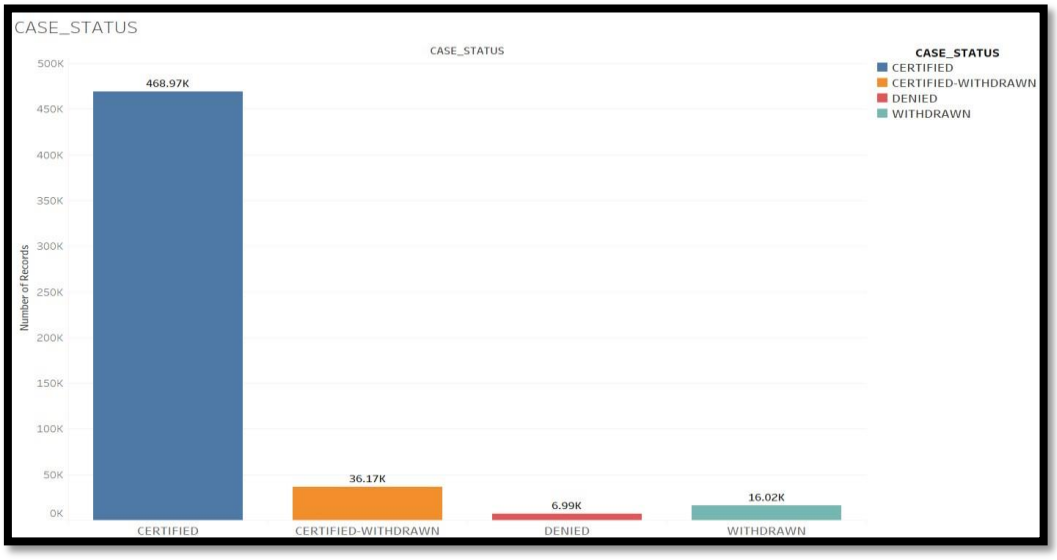

Figure 12: Case Status before transformation

B. 'TOTAL_WORKERS':

- Represents number of application filed by employer. **Figure13.** Maximum percentage of employer hire is 1 employee.

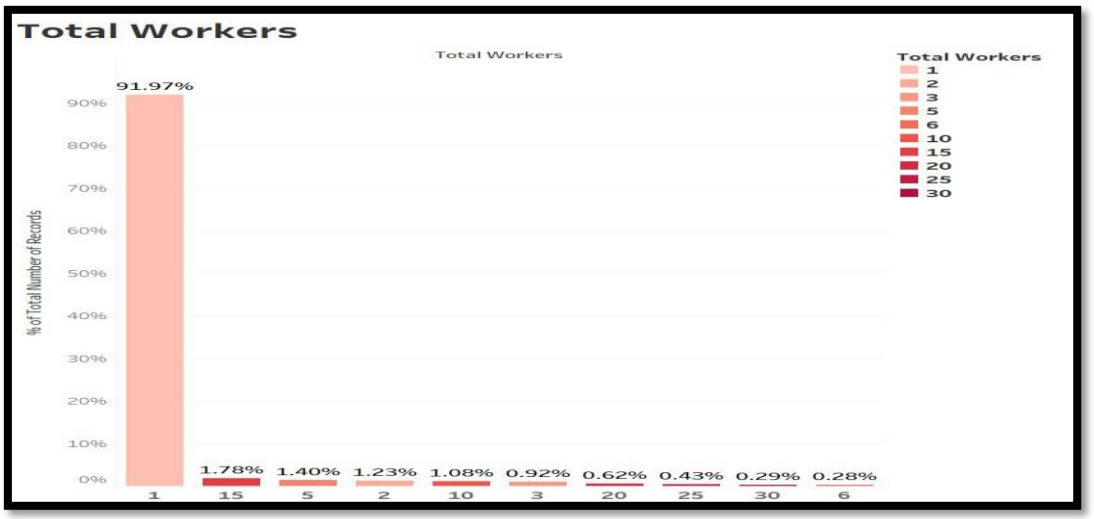

Figure 13: Total Workers hired by employer

C. 'FULL_TIME_POSITION':

- Positions are given in "Full Time Position = Y; Part Time Position = N" format. We converted them to dummies "Full Time Position_Y; Part Time Position_N" format". **Figure 14**

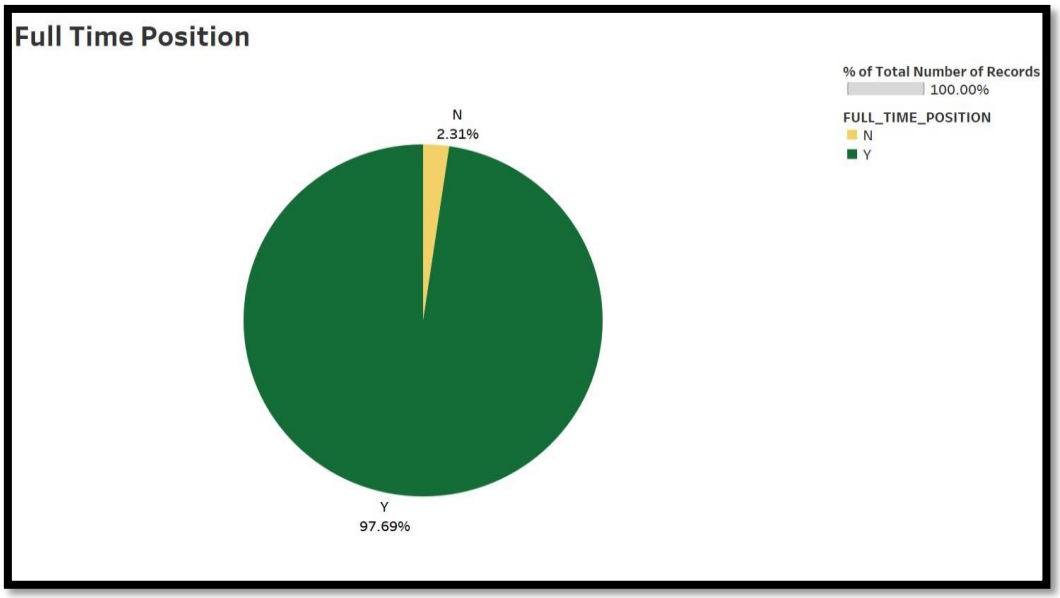

Figure 14: Full Time position before transformation

D. 'PREVAILING_WAGE':

- Prevailing wage is the average wage paid to employees with similar qualifications in the intended area of employment. **Figure 15** represent distribution prior transformation.

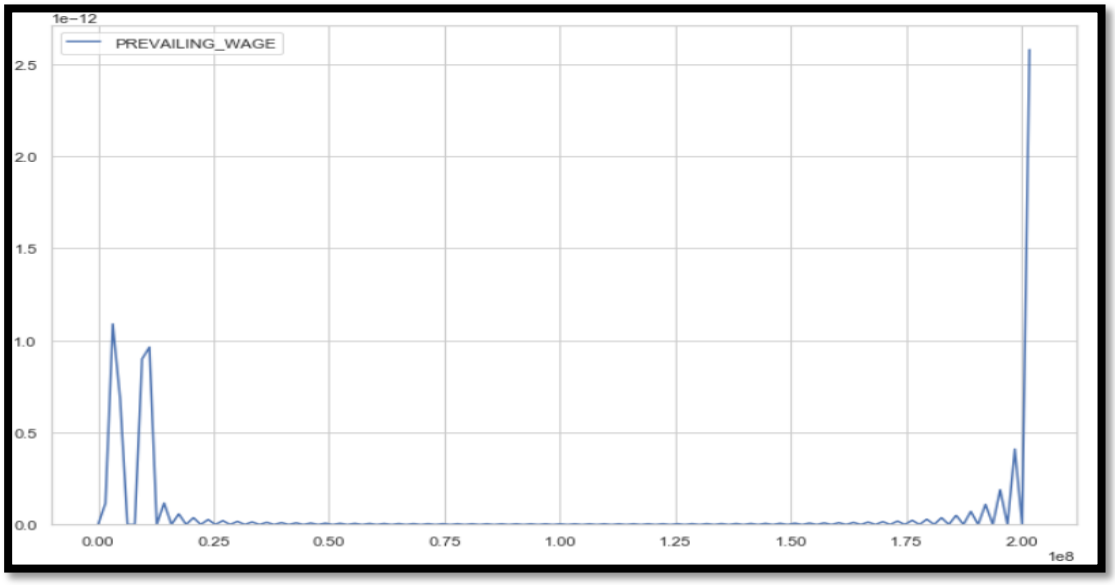

Figure 15: PREVAILING_WAGE density plot before transformation

- The *scipy.stats* library provides an implementation of the Box-Cox transform. We was able to transform PREVAILING_WAGE roughly to normal distribution by Box-Cox at confidence interval of 0.25 [8]. **Figure 16**

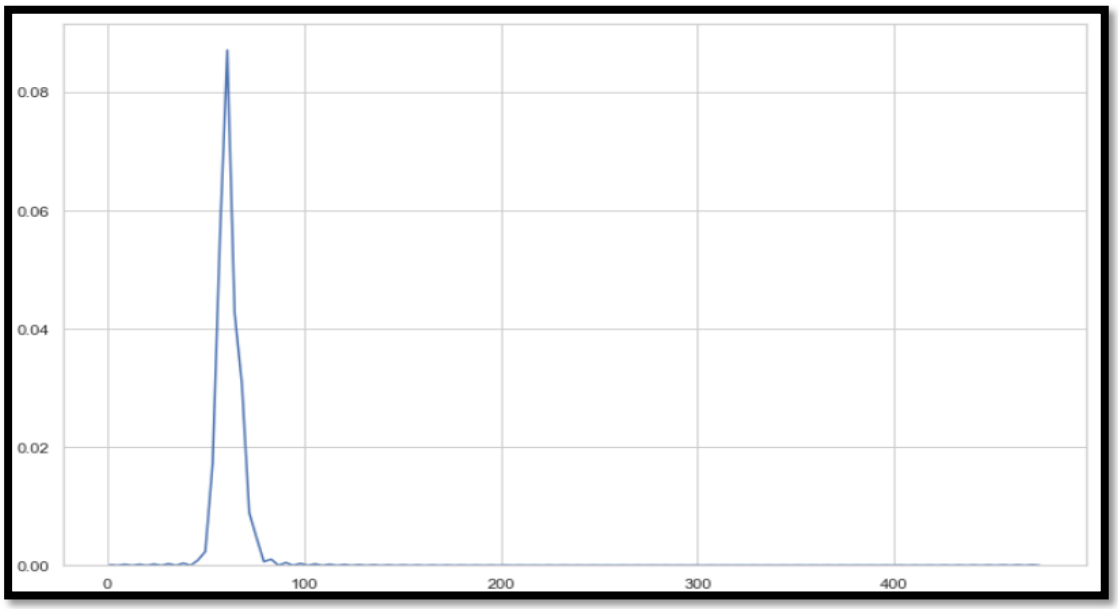

Figure 16: PREVAILING_WAGE density plot after Box-Cox transformation

E. 'PW_UNIT_OF_PAY':

- After transformation [refer section II.3.e.], year and hour was created and using one hot encoding converted into 0 and 1 respectively at final. **Figure 17**

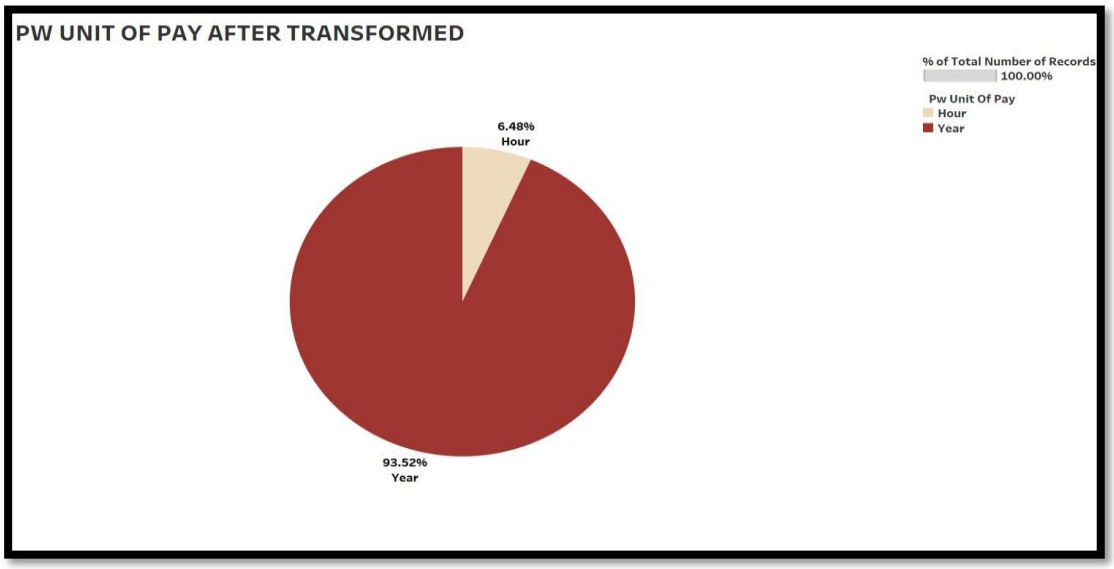

Figure 17: PW_UNIT_OF_PAY after Transformed

F. 'PW_SOURCE_YEAR':

- Year was extracted for which application was filed. We converted the data into one-hot representation at final step. **Figure 18**

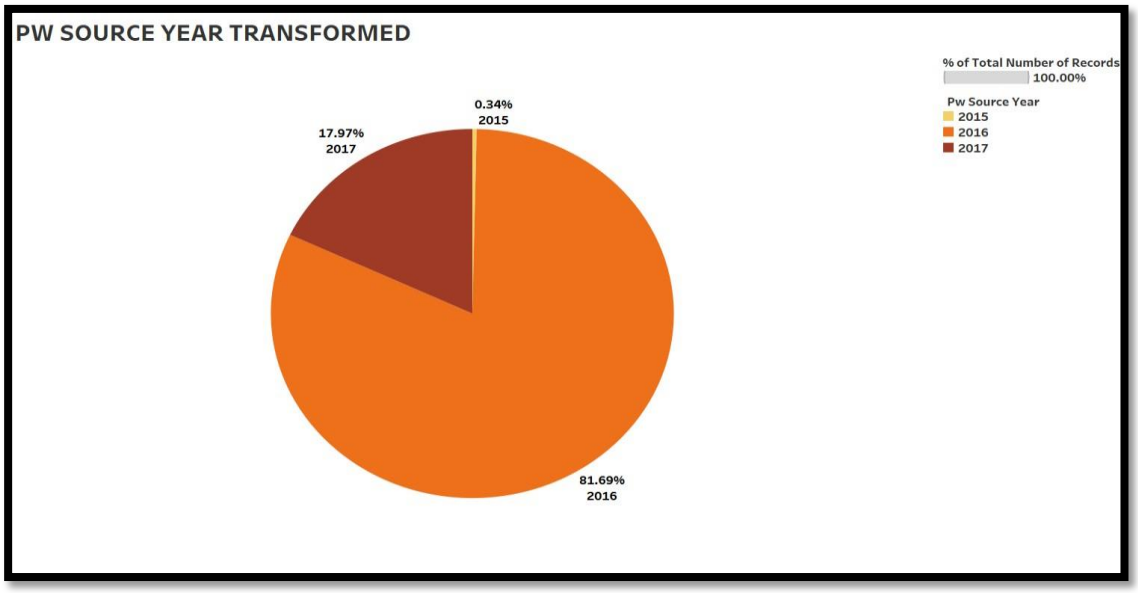

Figure 18: PW Source year after extraction of year

G. 'H-1B_DEPENDENT':

- H-1B-dependent employer is used by the United States Department of Labour to describe an employer who meets a particular threshold in terms of the fraction of the workforce comprising workers in H-1B status [9]. **Figure 19.** We have converted Y and N by one hot coding.

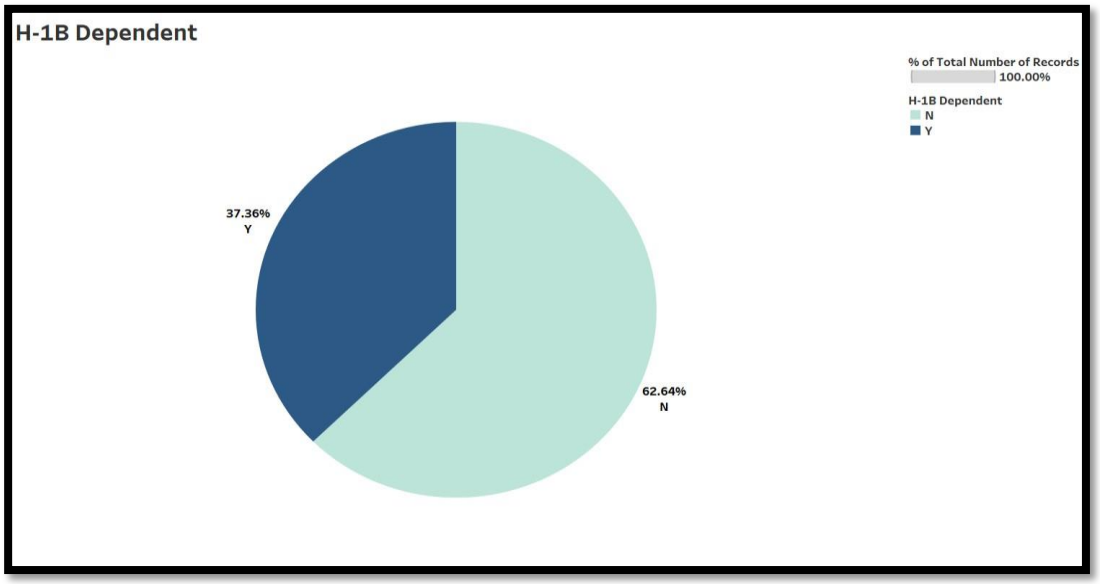

Figure 19: H-1B Dependent employer status

H. 'WILLFUL_VIOLATOR':

- Willful Violators or Willful Violator Employers are the employer who have committed either a willful failure or a misrepresentation of a material fact. A willful violator employer must comply with additional attestations under any LCA it files and are subject to random investigations by the Department of Labour within five years of the willful violation finding [10].
- As dataset hold most of employer who not wilfull violator. **Figure 20**

Willful Violator	
N	99.94%
Y	0.06%

Figure 20: Willful Violator ratio

- Transformed using by dummies i.e. one-hot encoding in final model.

I. 'Emp_Stay_Duration_Yr':

- Feature was extracted [refer section II.3.b]. It describe numbers of year applicant staying. In range of 0 to 3 year it was able to describe as with maximum percentage of 3 years duration of stay. **Figure 21**

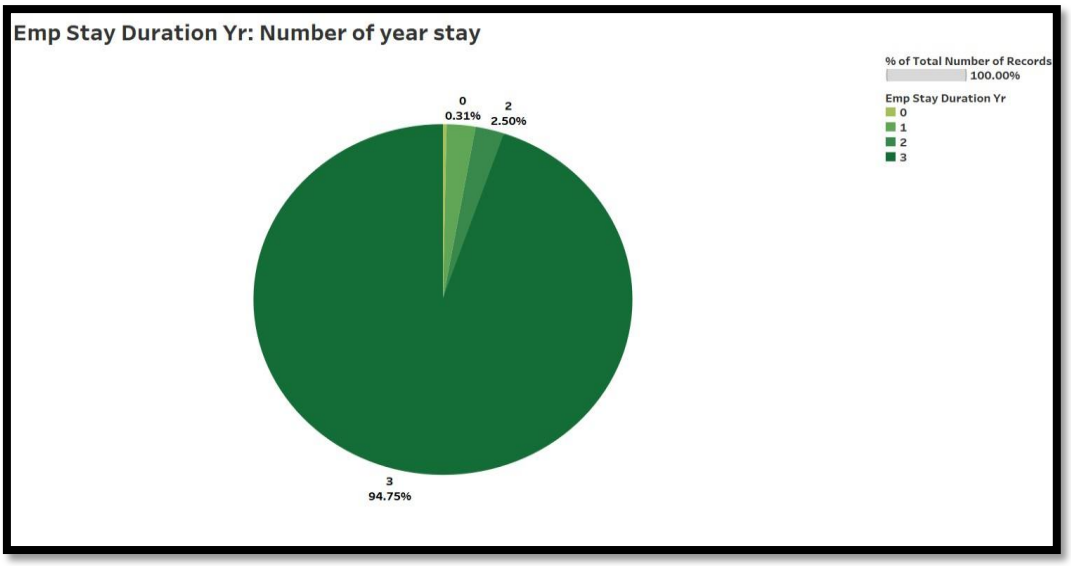

Figure 21: Emp Stay Duration in years

J. 'EMPLOYER_FREQUENCY':

- New Feature was added [Refer section II.3.f]. Describe frequent of employer in data.

K. 'AGENT_ATTORNEY': [Refer section II.3.g]

- Describes about Agent or Attorney filing an H-1B application on behalf of the employer with values 0 and 1

L. 'OCCUPATION':

- We created a feature [Refer section II.3.c]. Here applicant with computer occupation has maximum number of H-1B visa processing followed by architecture and engineering. **Figure 22**

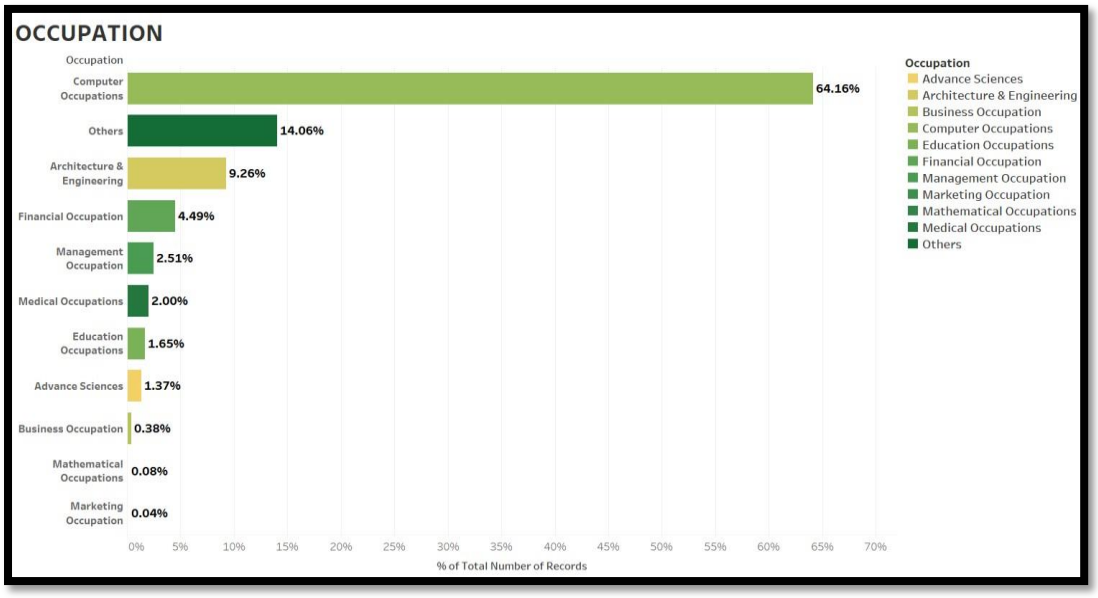

Figure 22: OCCUPATION

M. 'PWGrWGFm':

- Newly added feature [refer section II.3.e] **Figure 23** can be describe as follows**.** We have created dummy for by using one hot encoding.

Figure: 23: PWGRWGFR percent ratio

5. Summary of final feature: Table 1

Sr. No.	Features	Null value Treatment	Feature Transformed	Feature Extracted
1	**TOTAL_WORKER**	No	No	No
2	**PREVAILING_WAGE**	Yes	Yes	No
3	**EMPLOYER_FREQUENCY**	No	No	Yes
4	**FULL_TIME_POSITION**	No	Yes	No
5	**PW_UNIT_OF_PAY**	No	Yes	No
6	**H-1B_DEPENDENT**	Yes	Yes	No
7	**WILLFUL_VIOLATOR**	Yes	Yes	No
8	**DURATION_CS_DD**	No	No	Yes
9	**Emp_Stay_Duration_Yr**	No	No	Yes
10	**AGENT_ATTORNEY**	No	No	Yes
11	**Occupation**	No	No	Yes
12	**PWGrWGFm'**	No	No	Yes
13	**PW_SOURCE_YEAR**	No	Yes	No

Table 1: Summary of final Features

III. Statistical Analysis for Feature Selection

After features selection and extraction, final set of features from dataset **[section** II.4], we had applied statistical test like t-test and chi-square with respective to target variable to define which features are significant or not**.**

Most of data type of feature are of categorical (object) described and few numerical. **Figure 24.**

```
CASE_STATUS                 object
TOTAL_WORKERS                int64
FULL_TIME_POSITION          object
PREVAILING_WAGE            float64
PW_UNIT_OF_PAY              object
PW_SOURCE_YEAR             float64
H-1B_DEPENDENT              object
WILLFUL_VIOLATOR            object
CS_DD_Duration              object
Emp_Stay_Duration_Yr        object
EMPLOYER_FREQUENCY          object
AGENT_ATTORNEY              object
OCCUPATION                  object
PWGrWGFm                    object
dtype: object
```

Figure 24: Date type of features

Therefore we have applied only t-test and chi-square with respective to target feature 'CASE_STATUS'.

1. T_Test

A t-test is used to compare the mean of two given samples. Like a z-test, a t-test also assumes a normal distribution of the sample. A t-test is used when the population parameters (mean and standard validation) are not known.

There are three versions of t-test:

- Independent samples t-test which compares mean for two groups
- Paired sample t-test which compares means from the same group at different times
- One sample t-test which tests the mean of a single group against a known [**11**]

- **TOTAL_WORKER with 'CASE_STATUS':**

A. Visually **CASE_STATUS** with 1 i.e. CERTIFIED has maximum counts of TOTAL_WORKER. **Figure 25**

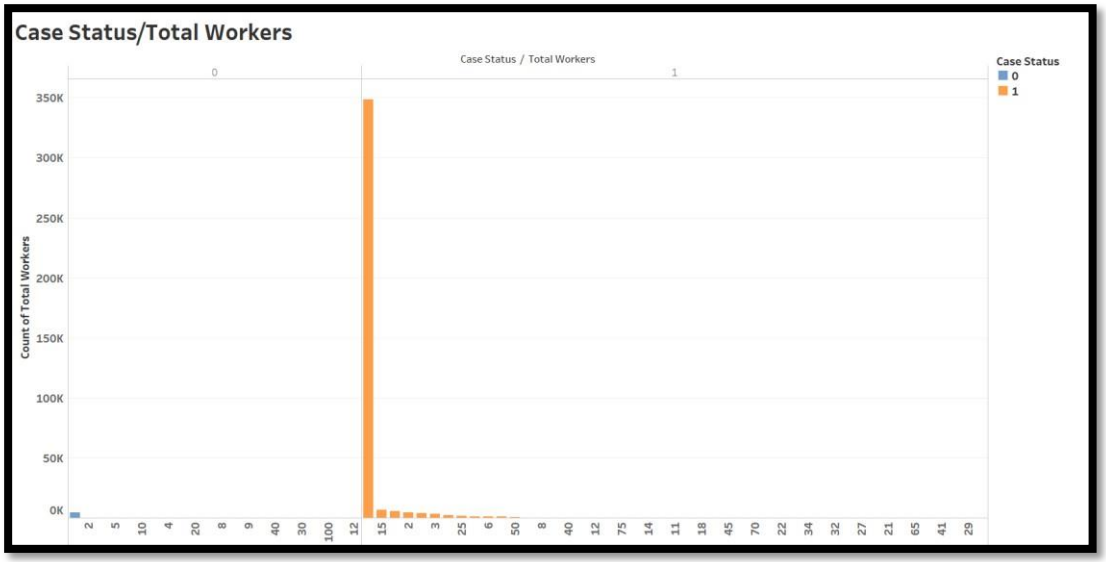

Figure 25: CASE_STATUS VS TOTAL_WORKER

B. Performed t-test as follows: **Figure 26**

- Defining **Null Hypothesis** and **Alternative Hypothesis**:

  - Null Hypothesis H0 = Mean count of **TOTAL_WORKER** for **CASE_STATUS** 0 and 1 are equal

  - Alternative Hypothesis H1= Mean count of **TOTAL_WORKER** for **CASE_STATUS** 0 and 1 are not equal

  - Using scipy stats library, t-test was calculated [12] **Figure 26**.

  - Considered level of significance of 5%. If P-value is greater than 0.05 of significance. We accept null hypothesis [13].

  - In our scenario we fails to reject null hypothesis. We got P-Value of 0.54.

  - Therefore **TOTAL_WORKERS** was **not statistically significant** with respective to **CASE_STATUS**.

```
a = df3[df3.CASE_STATUS == 1]['TOTAL_WORKERS']
b = df3[df3.CASE_STATUS == 0]['TOTAL_WORKERS']

from scipy.stats import ttest_ind

stat, p = ttest_ind(a, b)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('''Mean of Total workers with case status 0 and 1 are equal.
Therefore fails to reject H0.
Total Workers is not signifcant''')
else:
    print('''Mean of Total workers with case status 0 and 1 are equal.
Therefore Reject H0.
Total Workers is significant''')
```

```
stat=-0.601, p=0.548
Mean of Total workers with case status 0 and 1 are equal.
Therefore fails to reject H0.
Total Workers is not signifcant
```

Figure 26: T-Test for CASE_STATUS/TOTAL_WORKERS

- **PREVAILING_WAGE with CASE_STATUS: Figure: 27**

A. Mean prevailing wage with **CASE_STATUS** of 0 is more when compared with 1

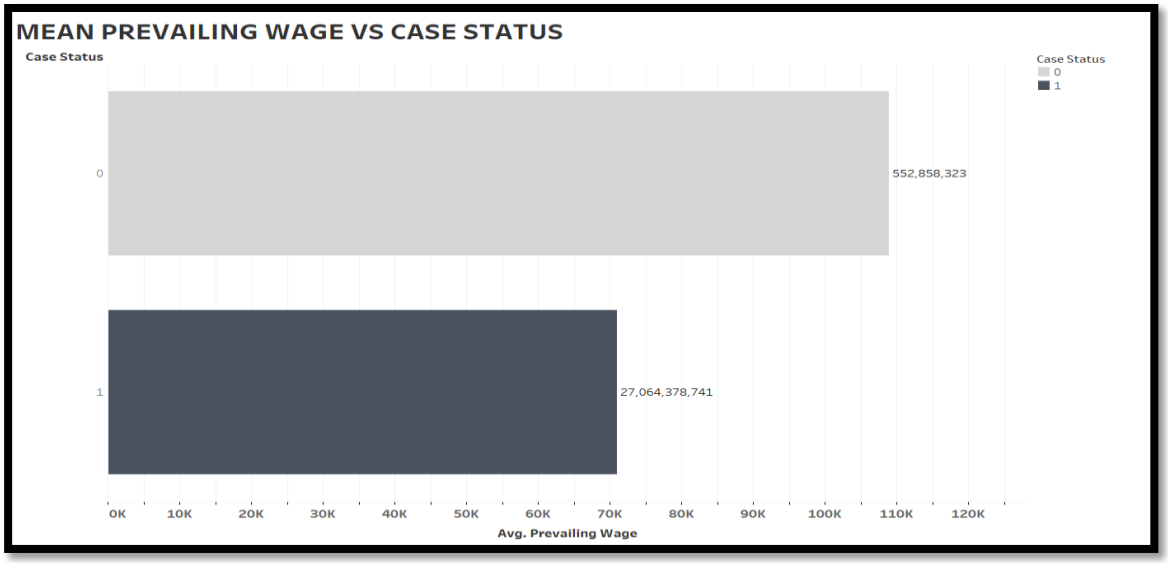

Figure 27: MEAN PREVAILING WAGE VS CASE STATUS

B. Similarly by using scipy.stat t-test, we concluded **PREVAILING_WAGE** was statistical significant with respect to **CASE_STATUS**.

- **EMPLOYER_FREQUENCY with CASE_STATUS: Figure: 28**

A. Mean **EMPLOPER_FREQUENCY** with **CASE_STATUS** with 1 is greater than compare to 0

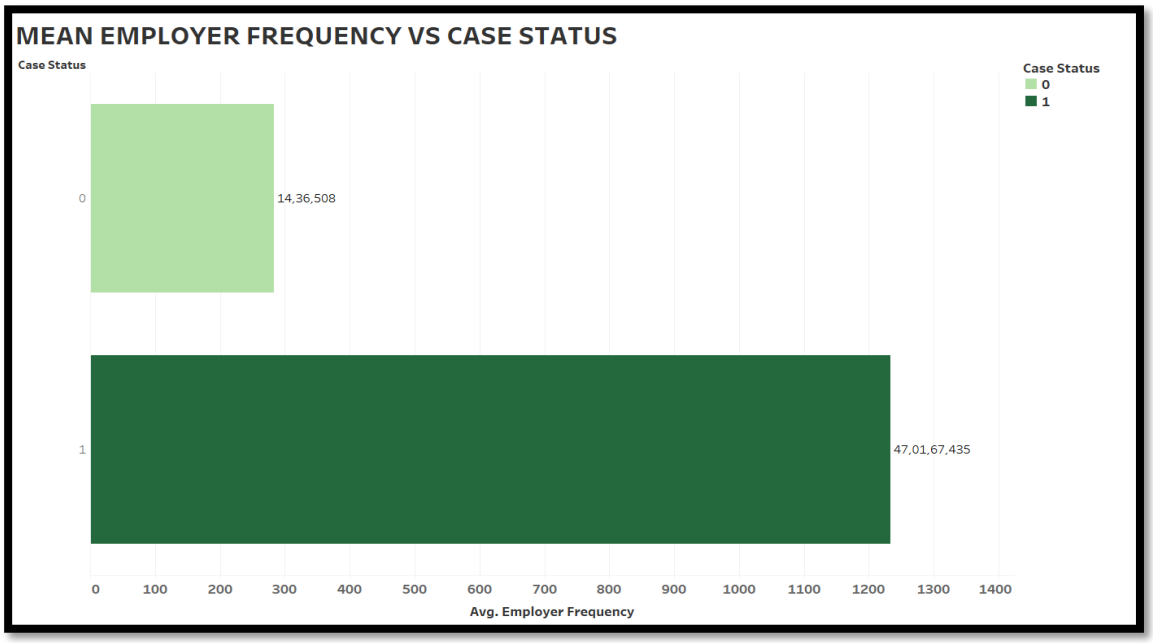

Figure 28: MEAN EMPLOYER_FREQUENCY VS CASE_STATUS

B. Statistically using t-test, we concluded **EMPLOYER_FREQUENCY** was statistically significant with **CASE_STATUS**.

2. Chi-Square

A Chi-square test is used in statistics to test the independence of two events. Given the data of two variables, we can get observed count O and expected count E. Chi-Square measures how expected count E and observed count O deviates each other.[14] **Figure 29**

The Formula for Chi Square Is

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

where:

c = degrees of freedom

O = observed value(s)

E = expected value(s)

Figure 29: Chi-square formula

Limitations: Chi-Square is sensitive to small frequencies in cells of tables. Generally, when the expected value in a cell of a table is less than 5, chi-square can lead to errors in conclusions.

Two types of chi-square tests:

- Chi-square goodness of fit test
- Chi-square test for independence

- **FULL_TIME_POSITION WITH CASE_STATUS:**

A. Performed chi-test as follows: **Figure 30**

B. Defining **Null Hypothesis** and **Alternative Hypothesis**:

- Null Hypothesis H0 = Observed values of **FULL_TIME_POSITION** and **CASE_STATUS** are same.
- Alternative Hypothesis H1= Observed values of **FULL_TIME_POSITION** and **CASE_STATUS** are not same.
- Using scipy stats library, chi-test was calculated [15].**Figure 30**
- Considered level of significance of 5%. If P-value is greater than 0.05 of significance. We accept null hypothesis.
- In this case we reject null hypothesis. We got P-Value of 0.00 less than level of significant.
- Therefore **FULL_TIME_POSTION** was **statistically significant** with respective to **CASE_STATUS**.

```
table = pd.crosstab(index=df3['CASE_STATUS'],columns=df3['FULL_TIME_POSITION'])
table
```

CASE_STATUS	FULL_TIME_POSITION N	Y
0	259	4817
1	9320	371995

```
from scipy.stats import chi2_contingency

stat, p, dof, expected = chi2_contingency(table)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('''Observerd values for Full_TIME_POSITTION and CASE_STATUS are same.
Therefore fails to reject H0.
FULL_TIME_POSTION are not signifcant''')
else:
    print('''Observed values for Full_TIME_POSITTION and CASE_STATUS are not same.
Therefore fails to reject H0.
FULL_TIME_POSTION are signifcant''')
```

```
stat=145.318, p=0.000
Observed values for Full_TIME_POSITTION and CASE_STATUS are not same.
Therefore fails to reject H0.
FULL_TIME_POSTION are signifcant
```

Figure 30: Chi-square test for FULL_TIME_POSITION and CASE_STATUS

- Similarly by using chi-square, features '**PW_UNIT_OF_PAY', 'H-1B_DEPENDENT', 'WILLFUL_VIOLATOR', 'DURATION_CS_DD', 'Emp_Stay_Duration_Yr', 'AGENT_ATTORNEY', 'Occupation','PWGrWGFm'** was found to be significant and features '**PW_SOURCE_YEAR'** was found to not significant.

3. Summary of Statistical Analysis: **Table 2**

Sr. No.	Features	Test	Significant
1	**TOTAL_WORKER**	t-test	No
2	**PREVAILING_WAGE**	t-test	Yes
3	**EMPLOYER_FREQUENCY**	t-test	Yes
4	**FULL_TIME_POSITION**	Chi-square	Yes
5	**PW_UNIT_OF_PAY**	Chi-square	Yes
6	**H-1B_DEPENDENT**	Chi-square	Yes
7	**WILLFUL_VIOLATOR**	Chi-square	Yes
8	**DURATION_CS_DD**	Chi-square	Yes
9	**Emp_Stay_Duration_Yr**	Chi-square	Yes
10	**AGENT_ATTORNEY**	Chi-square	Yes
11	**Occupation**	Chi-square	Yes
12	**PWGrWGFm'**	Chi-square	Yes
13	**PW_SOURCE_YEAR**	Chi-square	No

Table 2: Summary of Statistical Analysis

4. Statistical Significance Test comparison with Machine learning Algorithm:

Same was compared, feature importance which are statistical significant with feature importance extracted by random forest technique from base model after feature selection. **Figure 31**

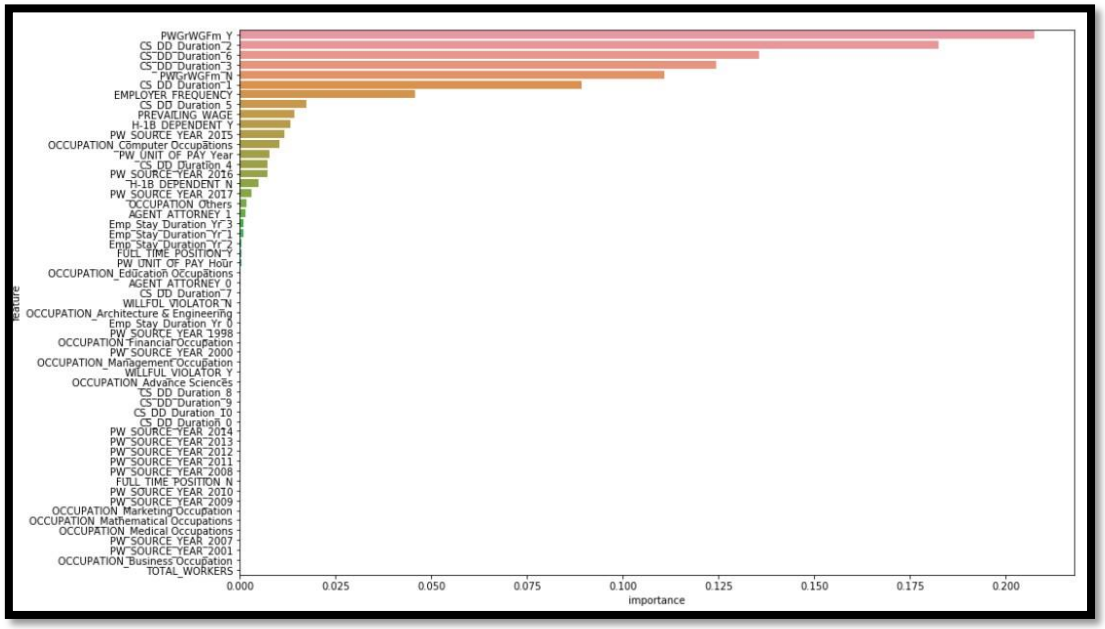

Figure 31: Feature importance by random forest technique from base model

IV. Model Building and Methods

1. Splitting data

- After the pre-processing steps described above, we split the training, validation and test sets 70:15:15.

- Due to the inherent bias in our dataset towards the "CERTIFIED" label, we created three versions of validation and test sets in order to make sense of the error analysis later on.

- First version of validation and test sets were both unbalanced.

- More specially, around 98% of the examples had a "CERTIFIED" label, mimicking the nature of the original dataset. **Figure 32**

```
(df["CASE_STATUS"].value_counts()/len(df["CASE_STATUS"])*100).round(2)

1    98.69
0     1.31
Name: CASE_STATUS, dtype: float64
```

Figure 32: Unbalanced target ratio

- Second version of validation and test sets were manually balanced by oversampling using SMOTE, "CERTIFIED" labelled examples roughly equal to the number of "DENIED" labelled examples.

- Third version of validation and test sets were under sampling using sampling technique.

2. One-hot encoding of features and Scaling

- After splitting data into X and y, dummies were created on all independent features and scaled by using min-max scalar prior splitting into train, validation and test.

- Total of 54 features were created by one hot encoding.

3. Classification models and techniques

- Our dataset is large, and it contains correlated features given the output. Also, the nature of the labelled features are categorical.

- Therefore, we thought supervised learning estimators like Logistic Regression, Naive Bayes, K Neighbors Classifier, and ensemble estimators like Decision TreeClassifier, Random Forest Classifier, would be the most feasible techniques for our application. Below, we provide brief descriptions for each of our techniques.

- Below represent overall summary and used of machine learning algorithms. **Figure 33**[15]

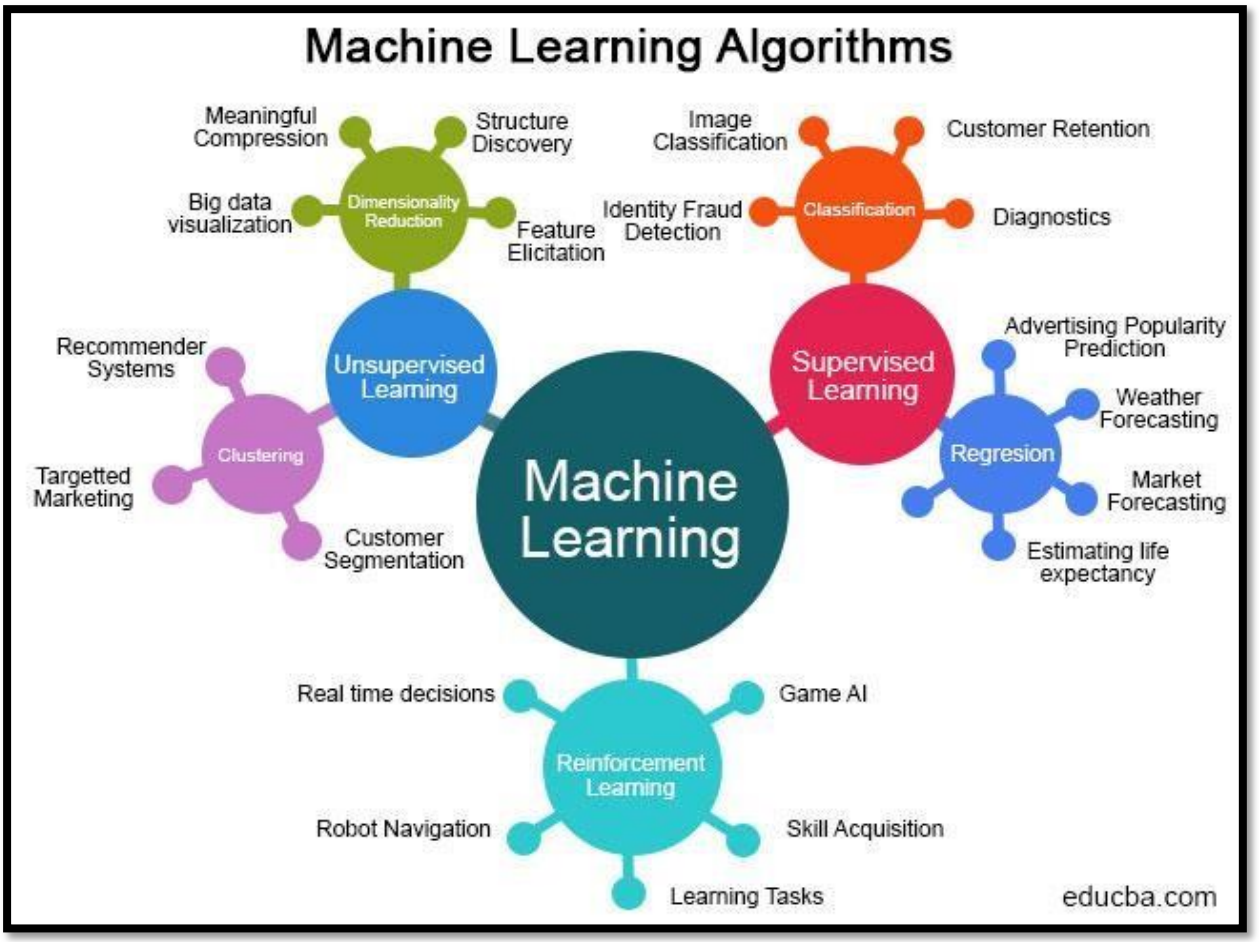

Figure 33: Machine Learning Algorithms summary

A. Logistic Regression

- Logistic regression is best suited for binary classification: data sets where y = 0 or 1, where 1 denotes the default class.
- Logistic regression is named after the transformation function it uses, which is called the logistic function.

$$\mathbf{h(x)= 1/ (1 + ex)}$$

- This forms a Sigmoid-shaped curve
- The goal of logistic regression is to use the training data to find the values of coefficients b0 and b1 such that it will minimize the error between the predicted outcome and the actual outcome. These coefficients are estimated using the technique of Maximum Likelihood Estimation.[16]

B. K-Nearest Neighbours

- K-nearest neighbours is a non-parametric machine learning model in which the model memorizes the training observation for classifying the unseen test data. It can also be called instance-based learning.
- This model is often termed as lazy learning, as it does not learn anything during the training phase like regression, random forest, and so on.
- The similarity between instances is calculated using measures such as Euclidean distance and Hamming distance.

C. Decision Tree

- A decision tree is drawn upside down with its root at the top.
- In the image on the left, the bold text in black represents a condition/internal node, based on which the tree splits into branches/ edges. The end of the branch that doesn't split anymore is the decision/leaf, in this case, whether the passenger died or survived, represented as red and green text respectively. **Figure 34**

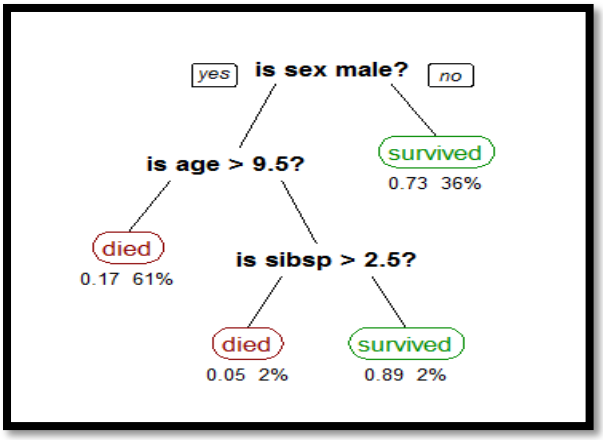

Figure 34: Decision Tree Flow

- A decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions.
- This algorithm is recursive in nature as the groups formed can be sub-divided using same strategy.
- Due to this procedure, this algorithm is also known as the greedy algorithm, as we have an excessive desire of lowering the cost. This makes the root node as best predictor/classifier.

D. Random Forest

- As an alternative to limiting the depth of the tree, which reduces variance (good) and increases bias (bad), we can combine many decision trees into a single ensemble model known as the random forest.

- The random forest is a model made up of many decision trees. Rather than just simply averaging the prediction of trees (which we could call a "forest"), this model uses two key concepts that gives it the name random:
    - Random sampling of training data points when building trees
    - Random subsets of features considered when splitting nodes
- The random forest combines hundreds or thousands of decision trees, trains each one on a slightly different set of the observations, splitting nodes in each tree considering a limited number of the features. The final predictions of the random forest are made by averaging the predictions of each individual tree

E. Gradient Boosting

- Boosting is a method of converting weak learners into strong learners. In boosting, each new tree is a fit on a modified version of the original data set.
- The gradient boosting algorithm (gbm) can be most easily explained by first introducing the AdaBoost Algorithm.
- The AdaBoost Algorithm begins by training a decision tree in which each observation is assigned an equal weight. After evaluating the first tree, we increase the weights of those observations that are difficult to classify and lower the weights for those that are easy to classify. The second tree is therefore grown on this weighted data.
- Gradient Boosting trains a number of models at a time in a gradual, additive and parallel manner.
- The major difference between AdaBoost and GradientBoosting is how the algorithms identify the shortcomings of weak learners, for e.g.: Decision Tree.

4. Summary for types of classification models which we applied on our dataset.

- By using Scikit-learn library were able to apply following models [17]. **Figure 35**

```
from sklearn.linear_model import LogisticRegression
lg = LogisticRegression()

from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier()

from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()

from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()

from sklearn.ensemble import GradientBoostingClassifier
gbm = GradientBoostingClassifier()

from sklearn.ensemble import BaggingClassifier
bg = BaggingClassifier()
```

Figure 35: Summary of supervised models applied for our dataset

V. Model Evaluation

1. Model Validation and Performance

- Model evaluation aims to estimate the generalization accuracy of a model on future data.
- Methods for evaluating a model's performance are divided into 2 categories: namely holdout and Cross-Validation. Both methods use a test set to evaluate model performance.[18]
- Holdout method: The purpose of holdout evaluation is to test a model on different data than it was trained on. In this method, the dataset is randomly divided into three subsets
  - Training set is a subset of the dataset used to build predictive models.
  - Validation set is a subset of the dataset used to assess the performance of the model built in the training phase. It provides a test platform for fine-tuning a model's parameters and selecting the best performing model. Not all modelling algorithms need a validation set.
  - Test set, or unseen data, is a subset of the dataset used to assess the likely future performance of a model. If a model fits to the training set much better than it fits the test set, overfitting is probably the cause.
- The holdout approach is useful because of its speed, simplicity, and flexibility. However, this technique is often associated with high variability since differences in the training and test dataset can result in meaningful differences in the estimate of accuracy.[18]

2. Hyperparameter Tuning Using GridSeachCV

- Machine learning algorithms have hyperparameters that allow you to tailor the behaviour of the algorithm to your specific dataset.
- This are different from parameters, which are the internal coefficients or weights for a model found by the learning algorithm. Unlike parameters, hyperparameters are specified by the practitioner when configuring the model.
- Typically, it is challenging to know what values to use for the hyperparameters of a given algorithm on a given dataset, therefore it is common to use random or grid search strategies for different hyperparameter values.
- Here, in case of our dataset we have used **GridSearchCV** to know the best parameters for our model. Before finding the best parameters, we built a model with ensemble techniques such as Random Forest, Decision Tree, Gradient Boosting, but all these models computed a low Precision, Recall, F1 score XXX.
- So after tuning the best parameters using GridSearchCV, we built a model with the same ensemble techniques that we used previously and we had a substantial increase Precision, Recall, F1 score after parameter tuning, to successfully predict the outcome of H-1B visa. **Figure 36**

```
params={"criterion":["gini","entropy"],"max_depth":[3,5,7],'n_estimators':[50,100],
       "min_samples_split":[2,5],'min_samples_leaf':[2,5]}
grid=GridSearchCV(estimator=rf,param_grid=params,cv=5)
grid.fit(xtrain_sampled,ytrain_sampled)
grid.best_params_
```

```
{'criterion': 'gini',
 'max_depth': 7,
 'min_samples_leaf': 5,
 'min_samples_split': 2,
 'n_estimators': 100}
```

```
rf_best=RandomForestClassifier(criterion= 'gini',max_depth= 7,min_samples_leaf= 5,min_samples_split= 2,n_estimators= 100)
```

Figure 36: Hyperparameter tuning for Random Forest

3. Classification Metrics

A. Confusion matrix:

A confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one.[19]

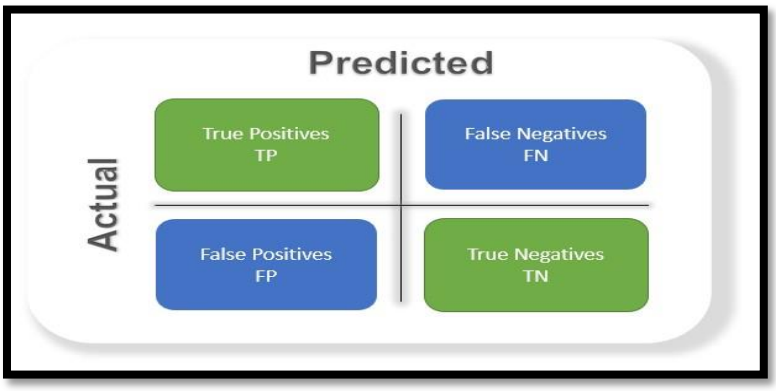

Figure 37: Confusion Matrix

B. Accuracy:

The most commonly used metric to judge a model and is actually not a clear indicator of the performance. The worse happens when classes are imbalance.

$$\frac{TP + TN}{TP + FP + TN + FN}$$

Figure 38: Accuracy Score

C. Precision:

Percentage of positive instances out of the total predicted positive instances. Here denominator is the model prediction done as positive from the whole given dataset.

$$\frac{TP}{TP + FP}$$

Figure 39: Precision Score

D. Recall/Sensitivity/True Positive Rate:

Percentage of positive instances out of the total actual positive instances. Therefore denominator (TP + FN) here is the actual number of positive instances present in the dataset.

$$\frac{TP}{TP + FN}$$

Figure 39: Recall Score

E. F1 Score:

It is the harmonic mean of precision and recall. This takes the contribution of both, so higher the F1 score, the better. See that due to the product in the numerator if one goes low, the final F1 score goes down significantly. So a model does well in F1 score if the positive predicted are actually positives (precision) and doesn't miss out on positives and predicts them negative (recall).[19

$$\frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 * precision * recall}{precision + recall}$$

Figure 40: F1 score

4. Summary, Comparison and Evaluation of models

- After feature model building, cross validation and hyperparameter tuning following model evaluation was performed as follows:

A. Models with Unbalanced target

As expected due to inherent bias in our dataset by target features having highly imbalance data for "CERTIFIED" i.e 1 and "DENIED i.e 0, overall score has impacted and keeping this as comparing bench mark with model build with Oversampling and Under sampling.

	Logistic Regression	Logistic Regression with threshold	Decision Tree	KNN3	KNN5	Random Forest	Bagging	Gradient Boosting
AUC Score	0.919617	0.919617	0.784339	0.7898	0.8043	0.832842	0.832745	0.927926
F1 Score	0.143800	0.798500	0.088600	0.1858	0.1904	0.460500	0.084200	0.191400
Precision	0.504800	0.820100	0.501300	0.5031	0.5028	0.512100	0.501800	0.503900
Recall	0.546500	0.779600	0.507300	0.5394	0.5363	0.672200	0.510000	0.551100
Test Accuracy	0.159300	0.159300	0.092800	0.2163	0.2229	0.758300	0.087800	0.224000
Train Accuracy	0.993400	0.993400	0.998500	0.9939	0.9930	0.998000	0.998000	0.993700
Validation Score	0.993378	0.993378	0.988546	0.9917	0.9917	0.991770	0.991873	0.993571

Figure 41: Evaluation for Unbalanced Models

B. Models with oversampling by SMOTE

SMOTE technique is used to deal with imbalanced dataset. SMOTE has significantly improved the precision and recall for all models. Of which Random forest hold overall best score when compared with other models and also cross validation score also was the best technique for assessing the effectiveness of the model built to accurately predict the outcome of H-1B visa applications.

	Logistic Regression	Decision Tree	Random Forest	Bagging	Gradient Boosting
F1 Score	0.871400	0.905000	0.907400	0.915000	0.923000
Test AUC Score	0.920037	0.778857	0.847631	0.838788	0.920238
Test Accuracy	0.917300	0.965000	0.965600	0.968000	NaN
Test F1 Score	0.576800	0.643800	0.653000	0.656900	0.611900
Test Precision	0.554900	0.600500	0.606600	0.611100	NaN
Test Recall	0.846000	0.778200	0.798300	0.786800	0.843300
Train Accuracy	0.853100	0.999200	0.996900	0.997200	0.898400
Validation Data AUC Score	0.933521	0.905572	0.962564	0.953583	NaN
Validation Data Accuracy	0.871700	0.905400	0.907800	0.915300	0.923000
Validation Data Precision	0.875100	0.912300	0.914300	0.921000	0.923900
Validation Data Recall	0.871700	0.905400	0.907800	0.915300	0.923000
Validation Score	0.853055	0.964812	0.966090	0.969228	0.895900

Figure 42: Evaluation for Oversampled Models

C. Models with Hyperparameter tuning

Model performed better after hyperparameter tuning and the predictive power of our model substantially increased as compared to the model built before hyperparameter tuning. This gave us average F1 Score, Recall, Precision Score overall when compared with model with imbalanced. The validation score that we obtained was a clear indication that overfitting problem of our model has been mitigated purely.

	Decision Tree	Random Forest	Bagging
F1 Score	0.850400	0.891000	0.740100
Test AUC Score	0.866669	0.922220	0.814759
Test Accuracy	0.859700	0.914100	0.782600
Test F1 Score	0.523800	0.573100	0.477300
Test Precision	0.531800	0.553000	0.518100
Test Recall	0.807800	0.845400	0.741200
Train Accuracy	0.826300	0.865500	0.732000
Validation Data AUC Score	0.910363	0.950242	0.816295
Validation Data Accuracy	0.850500	0.891000	0.740600
Validation Data Precision	0.850600	0.892200	0.742500
Validation Data Recall	0.850500	0.891000	0.740600
Validation Score	0.826455	0.867233	0.655121

Figure 43: Evaluation for Hyperparameter tune Models

D. Model with Under sampling

We consider using under sampling instead of oversampling in case of class imbalance in our Target variable, as we have 100K + records in your dataset. Oversampling is normally advisable only when we have class imbalance in < 10K samples data.

	Logistic Regression	Logistic Regression with threshold	Decision Tree	KNN	Random Forest	Bagging	Gradient Boosting
AUC Score	0.919956	0.919956	0.816680	0.879427	0.906778	0.906002	0.929593
F1 Score	0.844700	0.594500	0.814500	0.831100	0.837200	0.838100	0.853200
Precision	0.851900	0.759600	0.814500	0.832600	0.837200	0.838400	0.857200
Recall	0.845400	0.640800	0.814500	0.831300	0.837200	0.838100	0.853600
Test Accuracy	0.845400	0.845400	0.814500	0.831300	0.837200	0.838100	0.853600
Train Accuracy	0.855600	0.855600	0.996900	0.908000	0.986100	0.986900	0.876400
Validation Score	0.853503	0.853503	0.809604	0.831131	0.836337	0.839575	0.867574

5. Overall Conclusion for best method and model selection

- As we can conclude that models with under sampling gives best scores for F1, recall, accuracy when compared with models with oversampling and imbalanced model.

- Here **Random Forest** stands out best out of other models with under sampling, followed by bagging and gradient boosting classifier.

VI. Limitations, Conclusion and Future Work

---

1. Limitations/challenges

- The dataset was highly imbalanced and the reason for the model to not perform well enough was the imbalance nature of the dataset, given a balanced data would have performed really well.
- Also, this imbalanced nature of dataset affected Cohen kappa score, F1 score in the models that we built to know the accuracy.
- The highly imbalance nature of dataset could not be purely solved even by using techniques such as SMOTE (Oversampling, Under sampling)
- The original dataset before any EDA, Feature Transformation or SMOTE is highly biased towards predicting only a particular value, for eg: in the case status predicting More 1's (Certified) as compared to 0's (Denied).
- Also the data was provided was from a single financial year 2017.Getting more data from previous financial years would have provided us with rich quality of data to be dealt with.
- There are a lot of features created due to dummification. Extensive H-1B visa domain knowledge is required for selecting the features and limiting the dummified values.

2. Scope

Data for multiple years and with proper balance of acceptance and denial rate of H-1B visas will give better predictions and model performance.

3. CLOSING REFLECTIONS

Reflecting our learning from the project:

- SMOTE technique is used to deal with imbalanced dataset.
- To improve precision and recall in an imbalanced dataset, we must use ensemble techniques.
- User friendly codes have been used which are shorter, cleaner and more beautiful.

Reflecting our approach for consecutive projects:

- Continuous and Categorical variables need to be dealt separately and their interaction needs to be studied before we can further use them as features in a model.
- Based on the dataset, more stringent feature engineering and feature selection process needs to be done.
- Evaluating each step not only with a predictive point of view but also from a business perspective is equally important.

4. Conclusion and Future Work

- In the end, it is indeed possible to predict the outcomes of H-1B visa applications based on the attributes of the applicant using machine learning. Out of the models we tried**, Random Forest Classifier** with best parameters obtained using GridSearchCV outperformed all the other models with **98%** training accuracy and **86%** test accuracy on the under sampled balanced test data.
- That's likely because Random Forest Classifier are inherently better at explaining the complexities in the dataset. Overall, this model performed better after hyperparameter tuning and the predictive power of our model substantially increased as compared to the model built before hyperparameter tuning. This gave us a better F1 Score, Recall, Precision Score. The validation score that we obtained was a clear indication that overfitting problem of our model has been mitigated purely. This overfitting was caused due to the highly biased nature of the variables in our dataset. This cross validation score also was the best technique for assessing the effectiveness of the model built to accurately predict the outcome of H-1B visa applications.
- If we had more time and computational resources, there are several directions we could take to improve our prediction algorithm. First of all, we would try Random Forest with Lasso (L1 Norm) regularization, since we believe that some features are actually irrelevant to the output as previously discussed.
- Also, we could adjust the depth of the Random Forest, tune the hyperparameters a bit more precisely and possibly obtain the best model which would fairly predict the outcomes of H-1B visa applications and would give an optimal solution to the business problem favoured.
- In addition, we could convert more features such as **SOC NAME** into one-hot-k representation to achieve better accuracy. Finally, we could create more informative features such as Standard Industrial Classification codes of the companies through web crawling instead of using the given **EMPLOYER NAME** and **SOC NAME** features directly.

References

1. “H-1B Fiscal Year (FY) 2018 Cap Season,” USCIS. [Online]. Available: https://www. uscis.gov/working-united-states/temporary-workers/h-1b-specialty-occupationsand-fashion-models/h-1b-fiscal-year-fy-2018-cap-season.
2. “High-skilled visa applications hit record high,” CNNMoney. [Online]. Available: http: //money.cnn.com/2016/04/12/technology/H-1B-cap-visa-fy-2017/index.html.
3. “The H1B Process Explained, Step by Step”. [Online]. Available: https://www.stilt.com/blog/2018/08/h1b-process-explained-step-by-step/
4. “H-1B Disclosure Data FY17”. [Online].Available: https://data.world/ian/h-1-b-disclosure-data-fy-17
5. “Drop Columns with more than 60 Percent of “empty” Values in Pandas. [“Online”]. Available: https://stackoverflow.com/questions/49791246/drop-columns-with-more-than-60-percent-of-empt y-values-in-pandas.
6. “Prevailing Wage”. [“Online”]. Available:https://www.usavisanow.com/h-1b-visa/H-1B-visa-resources/prevailing-wage/
7. ”How Long an H-1B Worker Can Stay in the United States”. [“Online”]. Available: https://www.nolo.com/legal-encyclopedia/how-long-h-1b-worker-can-stay-the-united-states.html?
8. ” The Box-Cox Transformation”.[“Online”].Available: https://nickcdryan.com/2017/04/19/the-box-cox-transformation/
9. ”H-1B-dependent employer”. [“Online”].Available: https://en.wikipedia.org/wiki/H-1B-dependent_employer
10. ” H1B Visa Employer: Willful Violator”. [“Online”].Available: https://www.myvisajobs.com/H1B-Visa/Willful-Violator.aspx
11. “Statistical Tests — When to use which?”. [“Online”]. Available: https://towardsdatascience.com/statistical-tests-when-to-use-which-704557554740
12. ”Scipy.stats.ttestind”.[“Online”].Available:https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html
13. ” What a p-value tells you about statistical significance”. [“Online”]. Available: https://www.simplypsychology.org/p-value.html
14. ” Chi-Square Test for Feature Selection in Machine learning”. [“Online”]. Available: https://towardsdatascience.com/chi-square-test-for-feature-selection-in-machine-learning-206b1f0b8223
15. “scipy.stats.chi2_contingency”. [“Online”]. Available: https://docs.scipy.org/doc/scipy-0.15.1/reference/generated/scipy.stats.chi2_contingency.html
16. “Machine Learning Algorithms”. [“Online”]. Available: https://www.educba.com/machine-learning-algorithms/
17. ” Supervised learning”. [“ONLINE”] “https://scikitlearn.org/stable/supervised_learning.html”
18. “Machine Learning Algorithms”. [“Online”]. Available: https://machinelearningmastery.com/naive-bayes-for-machine-learning
19. ’’Model Evaluation Technique’’. [’’Online’’].Available: https://heartbeat.fritz.ai/introduction-to-machine-learning-model-evaluation-fa859e1b2d7f
20. “Machine Learning Algorithms”. [“Online”]. Available:https://towardsdatascience.com/various-ways-to-evaluate-a-machine-learning-models-performance-230449055f15