An Internship Program 2022

From 01-06-2022 to 15-07-2022

# Classification of Work Visa Approval using ML methods

supervised by

CEO& Founder: Yasin shah
Mentor: Karishma Kunwar

# Declaration

I hereby declare that the Internship submitted is our own unaided work. All direct or indirect sources used are acknowledged as references. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future Internship submitted. Further rights of reproduction and usage, however, are not granted here. This paper was not previously presented to another examination board and has not been published.

Signature
Team B

Table of Contents:

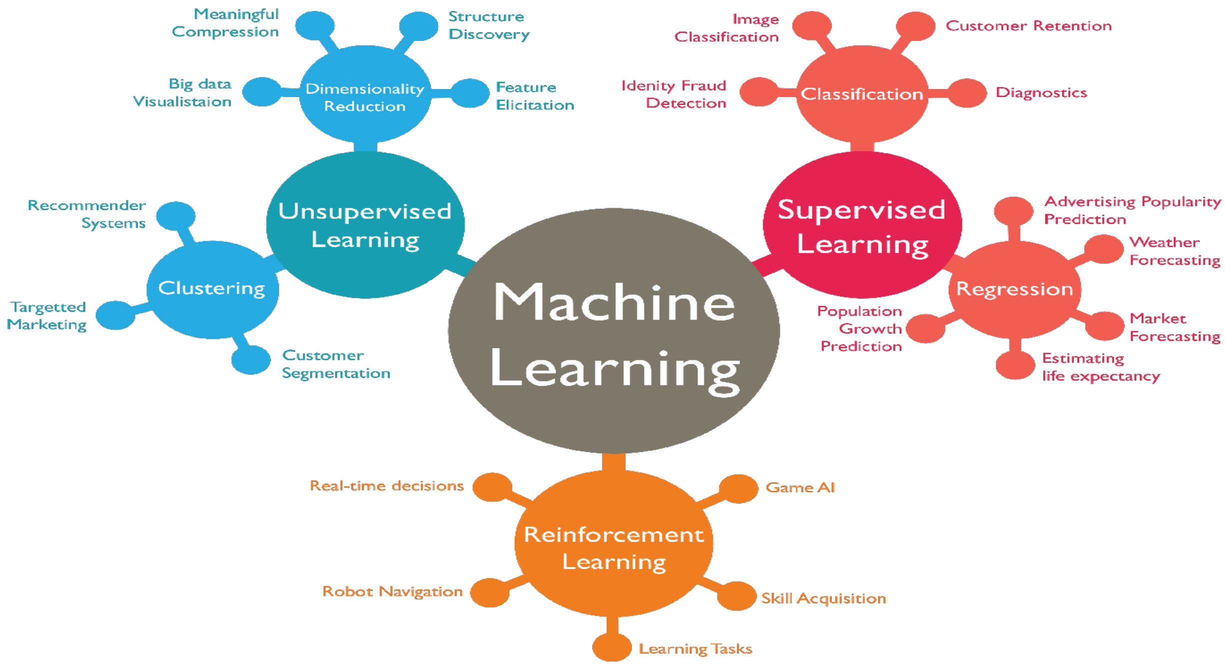
## Abstract:

H1B-Visa is the most sought-after non-immigrant visa that allows foreign workers to work in United States in speciality occupation. In 2019, more than 1 million applicants to get an H1B visa including new applications, renewals and transfer of H1-B to another company. There were more than 180,00 new applicants for H1-B, however, only 80,00 applications were picked up in the lottery process for taking it further to USCIS for approval.

The uncertainty in getting an H1-B visa creates employment and legal status uncertainties for a job application and high legal and visa processing fees for the organization over the period of employment. We plan to use the anonymized dataset for 2019 that United Status Department of Labor publishes publicly and apply data science techniques to improve predictability of approach.

# 1.Introduction:

Artificial intelligence and machine learning are among the most significant technological developments in recent history. Few fields promise to “disrupt” (to borrow a favored term) life as we know it quite like machine learning, but many of the applications of machine learning technology go unseen.
Machine learning is a subset of artificial intelligence in the field of the computer science that often uses statistical techniques to give computers the ability to “learn”.




Machine learning is a category of algorithm that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine leaning is to build algorithms that can receive input data and use statistical analysis to predict an output ehile updating outputs as new data becomes available.

The process involved in machine learning are similar to that of data mining and predictive modeling. Both require searching through data to look for patterns and adjusting program actions accordingly. Many people are familiar with machine leaning from shopping on the internet and being served ads related to their purchase. This happens because recommendation engines use machine learning to personalize online ad delivery in almost real time. Beyond personalized marketing, other common machine learning use cases include fraud detection, spam filtering, network security threat detection predictive maintenance and building news feeds.

The machine learning process includes the following steps:

1. Data Collection – The primary step of a Machine Learning process is gathering relevant information from various sources
2. Data Preparation – Once all the data is collected, it needs to be identified, sorted, and classified before analysing it. The techniques of data preparation depend on the kind of task that is to be done by the Machine Learning application.

3. Training – This stage involves training the machine to self-learn from the analysed data. Learning algorithms are created based on various parameters and expected outcomes of the application.
4. Evaluation – In this step, the Machine Learning application is tested to evaluate its performance and also identify bugs and find areas of improvement
5. Fine Tuning – Creating Machine Learning applications is a continuous process. As data preparation and analysing techniques evolve, the algorithms and the Machine Learning application model need to be fine-tuned.

H1-B visa data is taken from Kaggle. The csv file contains 589,414 attributes and 260 features. The first step towards solving any business problem using machine learning is hypothesis generation. Understanding the problem statement with good domain knowledge is important and formulating a hypothesis will further expose you to newer ideas of problem-solving. After hypothesis generation, 20 features has opted to implement data visualization and cleaning. In the following features has explained.

CASE_STATUS: Excluding the Withdrawn and Certified-Withdrawn, Certified decision is considered as 1 outcome in the resulting dataset and Denied as 0. This is used to model the outcome.

VISA_CLASS: Only H1-B visa class is being modeled in this paper which contributes to the majority of data points, we exclude the records for other work visas such as E-3 Australian, H-1B1 ChileandH-1B1Singapore.

EMPLOYER_NAME: The employer name submitting the visa application. We believe employer name is one of the important features to profile the visa application. As per NY times, some companies are manipulating the visa process by flooding the system.

SECONDARY_ENTITY: Whether the applicant will be places in a secondary location. This feature is assumed to be helpful since majority of Consultancy companies that are believed to outsource software services which have reputation to flood the visa processing.

AGENT_REPRESENTING_EMPLOYER: If another firm is representing the employer and its application. We plan to model this feature to see if an agency has high rejection rates as compared toanother.

JOB_TITLE, SOC_NAME: Job title and SOC name have details about the position, occupation field and seniority of the applicant.

SOC_CODE, NAICS_CODE: They are standard categories of a job.

CONTINUED_EMPLOYMENT: If this is a re-new visa application

CHANGE_PREVIOUS_EMPLOYMENT: If an application will continue without changes in job duties.

NEW_CONCURRENT_EMPLOYMENT: If the applicant will have an additional employer.

CHANGE_EMPLOYER: If applicant will get the visa with a new employer.

AMENDED_PETITION: If an applicant will work with the same employer with changes in duties.

FULL_TIME_POSITION: If this application is for full-time position

H-1B_DEPENDENT: If an employer is categorized to be H1-B dependent.

SUPPORT_H1B: If this application will be used in the future to file for H1-B petitions.

WILLFUL_VIOLATOR: If an employer has violated H1-B rules in the past.

WAGE_RATE_OF_PAY_FROM: Employer’s proposed wage rate.

WAGE_UNIT_OF_PAY: Paycheck frequency.

TOTAL_WORKER: Total amount of workers in the company filing the application.

In the below, the figure shows features and their data types.

```
Data columns (total 22 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   CASE_STATUS                   583806 non-null  object
 1   VISA_CLASS                    583806 non-null  object
 2   EMPLOYER_NAME                 583802 non-null  object
 3   AGENT_REPRESENTING_EMPLOYER   583801 non-null  object
 4   SECONDARY_ENTITY_1            534045 non-null  object
 5   JOB_TITLE                     583806 non-null  object
 6   SOC_TITLE                     583802 non-null  object
 7   SOC_CODE                      583802 non-null  object
 8   NAICS_CODE                    583805 non-null  float64
 9   CONTINUED_EMPLOYMENT          583806 non-null  object
 10  CHANGE_PREVIOUS_EMPLOYMENT    583806 non-null  int64
 11  NEW_CONCURRENT_EMPLOYMENT     583805 non-null  float64
 12  CHANGE_EMPLOYER               583806 non-null  int64
 13  AMENDED_PETITION              583806 non-null  int64
 14  H-1B_DEPENDENT                583786 non-null  object
 15  SUPPORT_H1B                   218528 non-null  object
 16  WILLFUL_VIOLATOR              583786 non-null  object
 17  WAGE_RATE_OF_PAY_FROM_1       583802 non-null  float64
 18  WAGE_RATE_OF_PAY_TO_1         298413 non-null  float64
 19  WAGE_UNIT_OF_PAY_1            583802 non-null  object
 20  TOTAL_WORKER_POSITIONS        583805 non-null  float64
 21  PREVAILING_WAGE_1             579125 non-null  float64
dtypes: float64(6), int64(3), object(13)
```

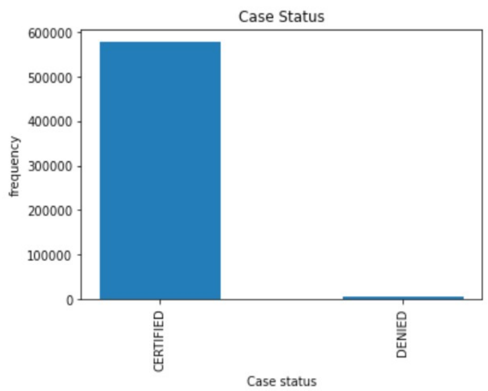
## 2. Exploratory Data Analysis:

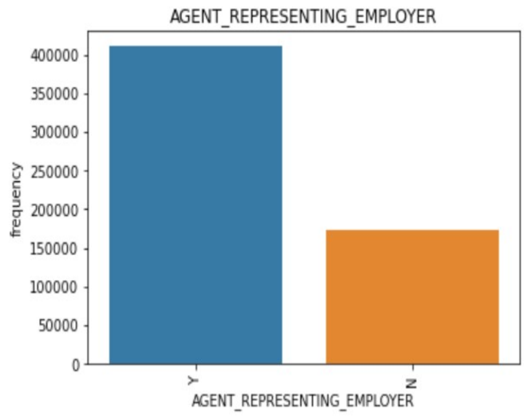
In VISA_CLASS, It has focusing H1B Visa for united states of America. So, drop the rows which does not belongs to H1B visa. In CASE_STATUS is target feature and has consists of Certified, Withdrawn, Certified Withdrawn and Denied. Here, considering the applicant who were Certified or Denied. So,  Withdrawn and Certified Withdrawn is dropped out from the list. In the following the figure shows code of execution.

```
H1B_visa = H1B_visa[H1B_visa.VISA_CLASS == 'H-1B']
H1B_visa = H1B_visa[H1B_visa.EMPLOYER_COUNTRY == 'UNITED STATES OF AMERICA']
H1B_visa = H1B_visa[H1B_visa.CASE_STATUS != 'WITHDRAWN']
H1B_visa = H1B_visa[H1B_visa.CASE_STATUS != 'CERTIFIED-WITHDRAWN']
```

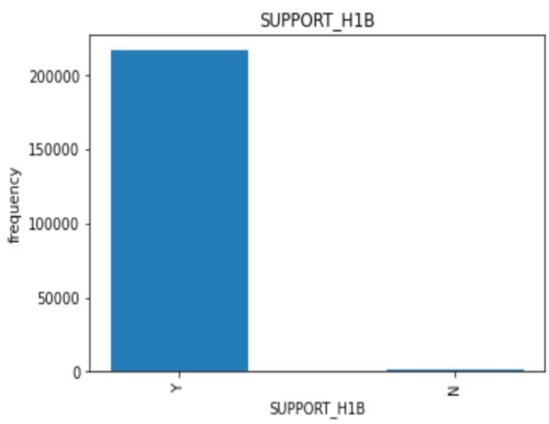
## 2a. Data Visualization:

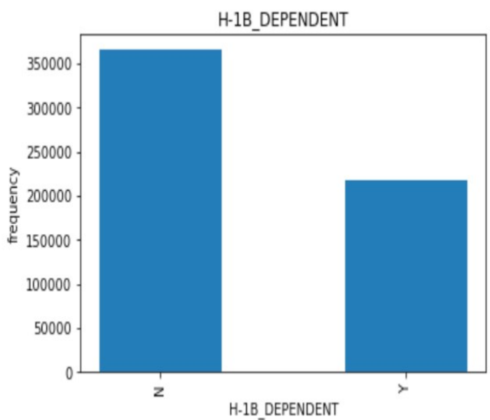
In CASE_STATUS is plotted as bar plot and it is observed that target feature is biased. In the modelling part, further explanation is provided. On other side, AGENT_REPRESENTI-NG_EMPLOYER has enough Y or N which doesn't comes under biased feature.



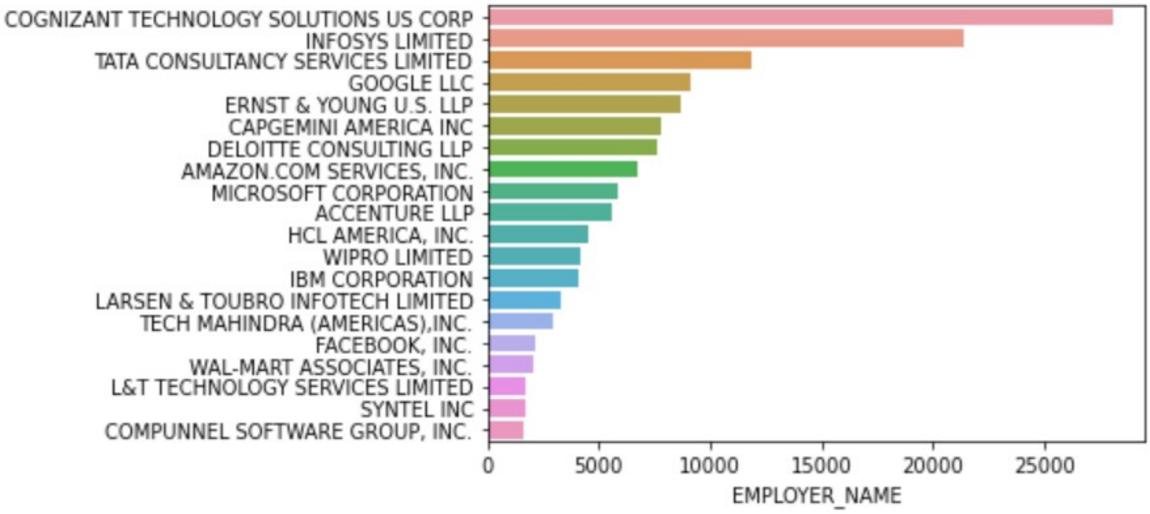




SUPPORT__H1B, H1B-DEPENDENT Features have binomial variables which is like Y or N is plotted here in the beneath.



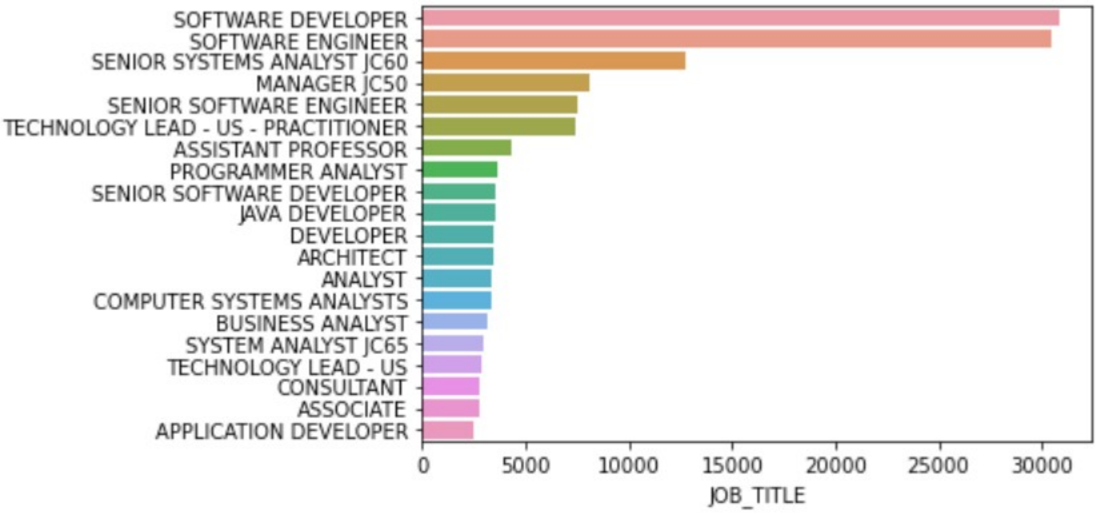



In Features, it has both numerical and categorical  attributes. But, Some of the features has large set of categorical attributes or entities. In order to reduce the large of categorical variables, bin counting

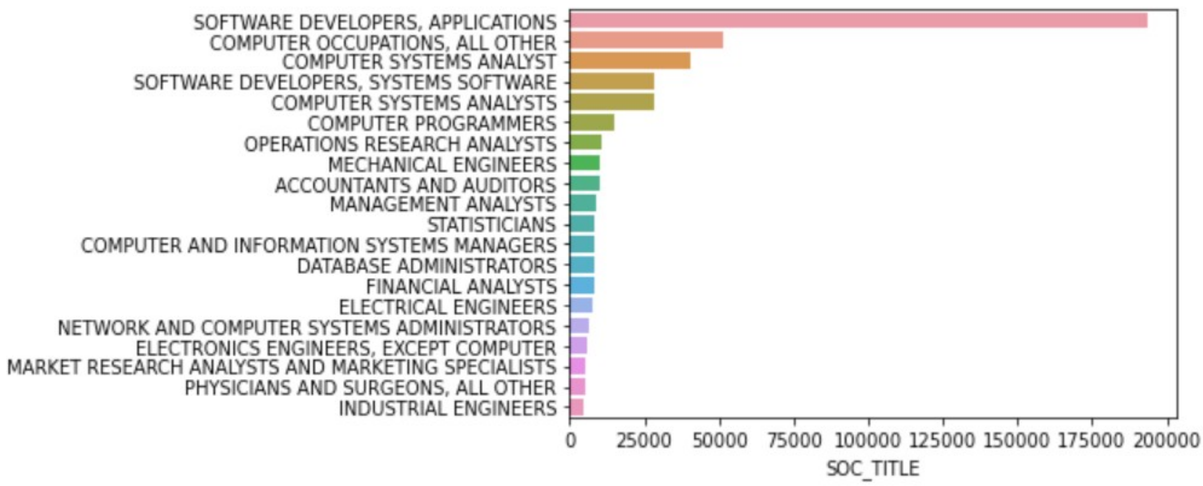



is introduced to reduce the computational cost. EMPLOYEE_NAME, SOC_TITLE, SOC_CODE, JOB_TITLE, and NAICS CODE are comprised under the large of categorical variables.

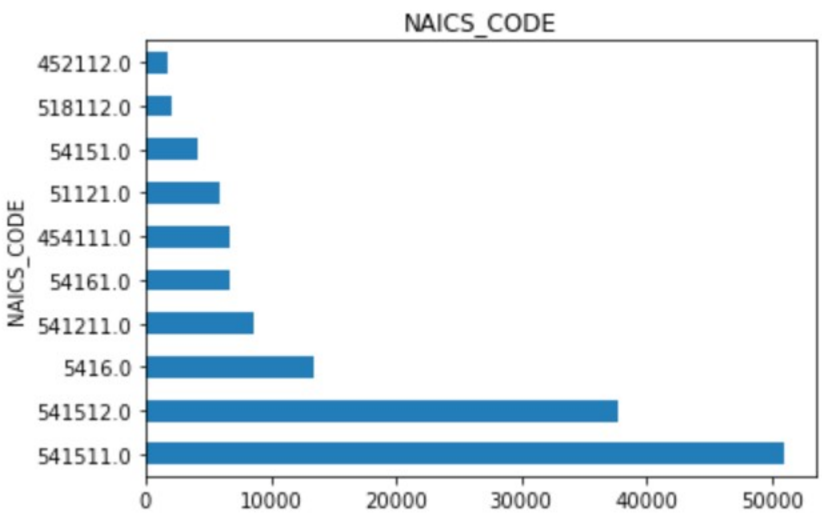
In the following figures, above categorical features is plotted with the top 20 features.
From the above Figure, It is clearly visible that IT Service based employees are the major applicant of H1-B Visa.




JOB_TITLE Figure, shows that software developer and software engineer are highly recruited around the USA and the third position took the seniority level position.




SOC_TITLE reveals the information that software has biased with core field such as medical, engineering science, marketing, network engineers and so on.




In NAICS_CODE, According to US Visa applicant has specific identity number for each job role and considering the number could be state level and regional companies or it could be anything. 541511 NAICS_CODE has highest number of applicants.

So far, it is depicted feature visualization of all proposed features except, numerical features.

## 2b. Data Cleaning:

In this subsection, data cleaning along with data visualization of some of the remaining features will be shown here.

```
H1B_visa.isnull().sum()
```

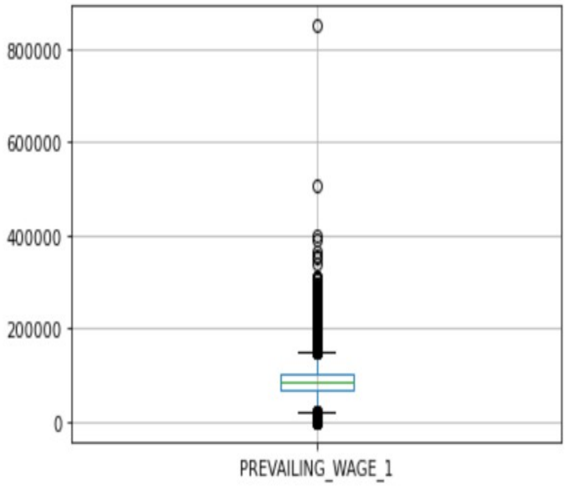
```
CASE_STATUS                          0
VISA_CLASS                           0
EMPLOYER_NAME                        4
AGENT_REPRESENTING_EMPLOYER          5
SECONDARY_ENTITY_1               49761
JOB_TITLE                            0
SOC_TITLE                            4
SOC_CODE                             4
NAICS_CODE                           1
CONTINUED_EMPLOYMENT                 0
CHANGE_PREVIOUS_EMPLOYMENT           0
NEW_CONCURRENT_EMPLOYMENT            1
CHANGE_EMPLOYER                      0
AMENDED_PETITION                     0
H-1B_DEPENDENT                      20
SUPPORT_H1B                     365278
WILLFUL_VIOLATOR                    20
WAGE_RATE_OF_PAY_FROM_1              4
WAGE_RATE_OF_PAY_TO_1           285393
WAGE_UNIT_OF_PAY_1                   4
TOTAL_WORKER_POSITIONS               1
```

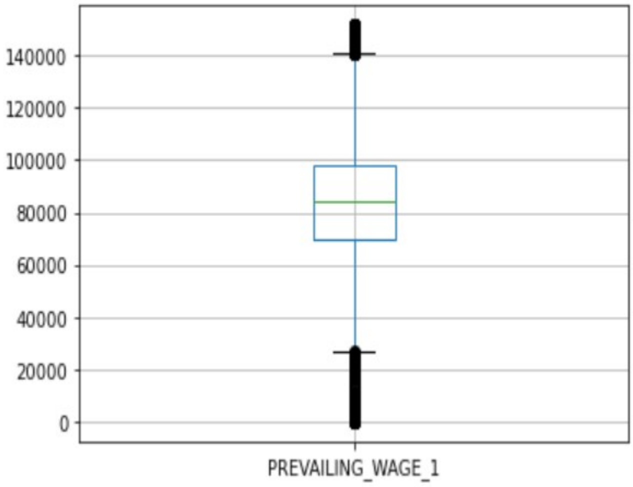
EMPLOYER_NAME, AGENT_REPRESENTING_EMPLOYER, SOC_TITLE, SOC_CODE, NAICS_CODE, NEW_CONCURRENT_EMPLOYMENT, H-1B_DEPENDENT, WILLFUL _VIOLATOR, WAGE_RATE_OF_PAY_FROM_1, WAGE_UNIT_OF_PAY_1, and TOTAL _WORKER_ POSITIONS are very less missing variables. But, It would be needed sometimes to increase the accuracy rate.

In the below graph, SECONDARY_ENTITY_1,SUPPORT_H1B, WAGE_RATE_OF_PAY_TO_1 will be filled or replaced with mode or median and mean.

```
H1B_visa['EMPLOYER_NAME'] = H1B_visa['EMPLOYER_NAME'].fillna(H1B_visa['EMPLOYER_NAME'].mode()[0])
H1B_visa['AGENT_REPRESENTING_EMPLOYER'] = H1B_visa['AGENT_REPRESENTING_EMPLOYER'].fillna(H1B_visa['AGENT_REPRESENTING_EMPLOYER'].
H1B_visa['SECONDARY_ENTITY_1'] = H1B_visa['SECONDARY_ENTITY_1'].fillna(H1B_visa['SECONDARY_ENTITY_1'].mode()[0])
H1B_visa['SOC_CODE'] = H1B_visa['SOC_CODE'].fillna(H1B_visa['SOC_CODE'].mode()[0])
H1B_visa['NAICS_CODE'] = H1B_visa['NAICS_CODE'].fillna(H1B_visa['NAICS_CODE'].mode()[0])
H1B_visa['SOC_TITLE'] = H1B_visa['SOC_TITLE'].fillna(H1B_visa['SOC_TITLE'].mode()[0])
H1B_visa['H-1B_DEPENDENT'] = H1B_visa['H-1B_DEPENDENT'].fillna(H1B_visa['H-1B_DEPENDENT'].mode()[0])
H1B_visa['WILLFUL_VIOLATOR'] = H1B_visa['WILLFUL_VIOLATOR'].fillna(H1B_visa['WILLFUL_VIOLATOR'].mode()[0])
H1B_visa['NEW_CONCURRENT_EMPLOYMENT'] = H1B_visa['NEW_CONCURRENT_EMPLOYMENT'].fillna(H1B_visa['NEW_CONCURRENT_EMPLOYMENT'].mode(
H1B_visa['WAGE_RATE_OF_PAY_FROM_1'] = H1B_visa['WAGE_RATE_OF_PAY_FROM_1'].fillna(H1B_visa['WAGE_RATE_OF_PAY_FROM_1'].mode()[0])
H1B_visa['WAGE_UNIT_OF_PAY_1'] = H1B_visa['WAGE_UNIT_OF_PAY_1'].fillna(H1B_visa['WAGE_UNIT_OF_PAY_1'].mode()[0])
H1B_visa['TOTAL_WORKER_POSITIONS'] = H1B_visa['TOTAL_WORKER_POSITIONS'].fillna(H1B_visa['TOTAL_WORKER_POSITIONS'].mode()[0])
```

For wage rate scale, PREVAILING_WAGE_1  has outlier and applied quantile. In the down, Before and After Quantile is plotted.



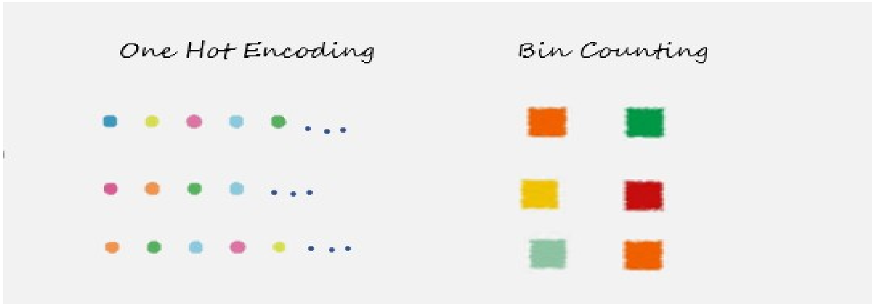

WAGE_RATE_OF_PAY_TO_1 is followed the quantile procedure and eliminated the outlier and filled with 0.25 and 0.9 quantile percentages.

## 3. Feature Engineering Analysis

In this section, large set of Categorical data is converted low number of Categorical variables or attributes using bin counting. Further, It has to be converted into numerical variables with help of Encoding techniques will be discussed in the following subsection of the section 3.

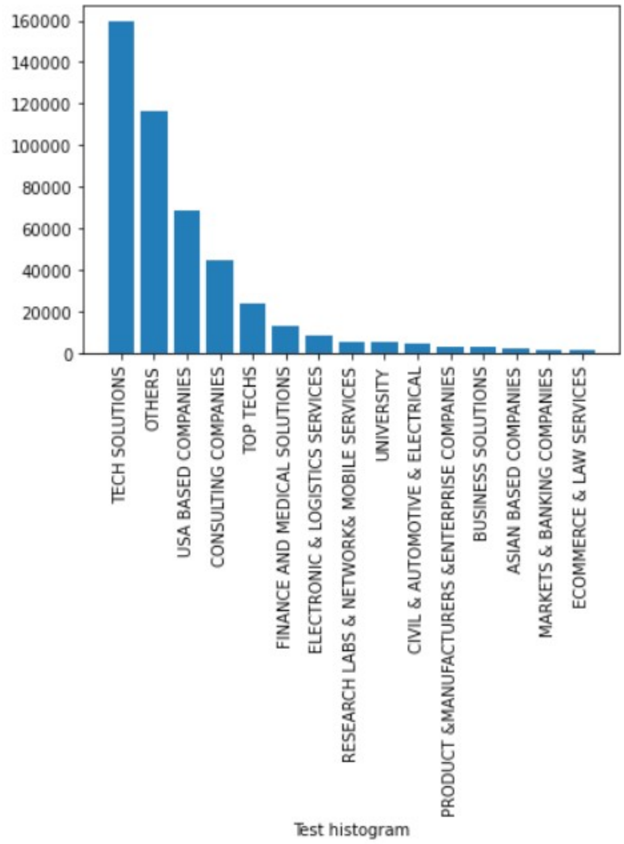
### 3(i). Bin counting(Response encoding):

The idea of bin counting is deviously simple: rather than using the value of the categorical variable as the feature, use the conditional probability of the target under that value.

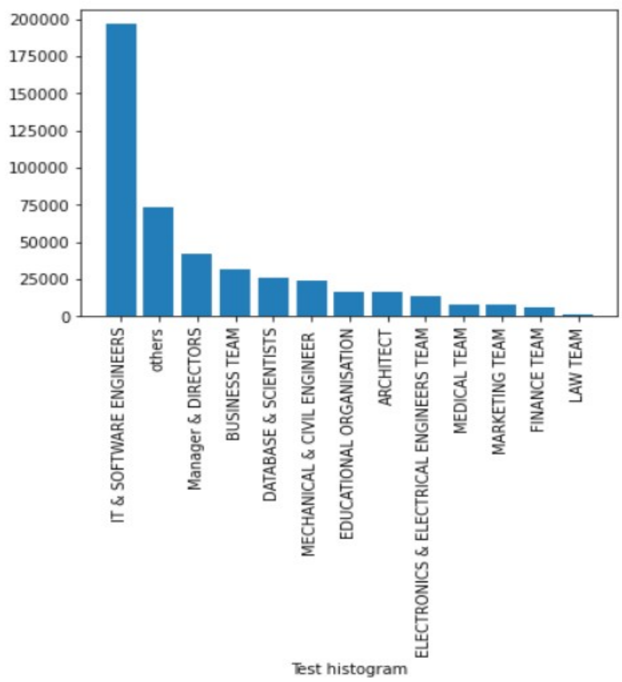



EMPLOYEE_NAME, SOC_TITLE, SOC_CODE, JOB_TITLE, and NAICS CODE are converted into bins using field as conditional probability.

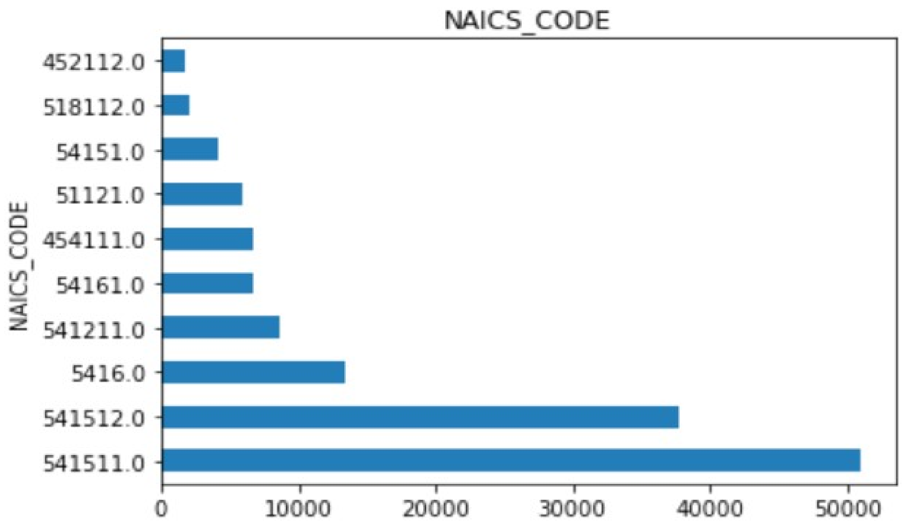
TOP TECHS, ELECTRONIC & LOGISTICS SERVICES, E-COMMERCE & LAW SERVICES, UNIVERSITY, MARKETS & BANKING COMPANIES, BUSINESS SOLUTIONS, FINANCE AND MEDICAL SOLUTIONS, RESEARCH LABS & NETWORK& MOBILE SERVICES, TECH SOLUTIONS, CONSULTING COMPANIES, USA BASED COMPANIES, PRODUCT &MANUFACTURERS &ENTERPRISE COMPANIES, and CIVIL & AUTOMOTIVE & ELECTRICAL are these considered as new categorical values as EMPLOYER BRANCH.




SOC_TITLE, SOC_CODE, JOB_TITLE, and NAICS CODE is proceed with same process with new categorical values. Both SOC_TITLE and JOB_TITLE are converted into the SOC_TITLE_NEW and JOB_TITLE_NEW.




```
IT ENGINEERS                 294574
DATABASE & SCIENTISTS         28724
MANAGER                       23598
ELECTRONICS ^ LOGISTICS       22647
MECHANICAL & CIVIL            18571
Education                     13740
FINANCE                       12576
MEDICAL                       11183
AUDIT & ADVERTISEMENT          9653
SALES & EXECUTIVES             8548
TECHNICIANS                    7241
H.R & FASHION                  4776
AGRICULTURE & CHEFS            1863
ADMINSTRATIVE & LAW            1510
P.R & URBAN                    1370
EDUCATION & TRAINING           1291
CHEMICAL ENGINEERS             1043
Name: SOC_TITLE_NEW, dtype: int64
```




NAICS_CODE is shown in the side figure before converting into low bin counting. First two digit numbers are used an Unicode as number in conditional statement.

### 3(ii). Categorical Encoding:

Converting the data types of string and object is converted into numerical data types which is integer. It has categorized into the following types: a)Label Encoding, b) One-hot Encoding and c) Target Encoding. Even though many classification developed, but only a few are mentioned here.

For this model, Label Encoder is used to convert into numerical attributes. EMPLOYEE_NAME, SOC_TITLE, SOC_CODE, JOB_TITLE, CONTINUED_EMPLOYMENT and NAICS CODE are transformed into numerics which is from 0 to 14 (which varies depend on the feature).

The features has attributes as Y or N that means either Yes or No, which are converted into 0 and 1 using Label Encoder. In the below figure, CASE_STATUS_N is used transformed with Label Encoder.

CASE_STATUS of CERTIFIED = 0, CASE_STATUS of DENIED = 1

```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
le.fit(H1B_visa.CASE_STATUS)
# print list(le.classes_)
H1B_visa['CASE_STATUS_N']=le.transform(H1B_visa['CASE_STATUS'])
H1B_visa['CASE_STATUS_N'].value_counts()
```

```
0    458192
1      4716
Name: CASE_STATUS_N, dtype: int64
```

Remaining all features are converted into numericals with Label Encoder.

```
le = preprocessing.LabelEncoder()
le.fit(H1B_visa.SECONDARY_ENTITY_1)
# print list(le.classes_)
H1B_visa['SECONDARY_ENTITY_1_N']=le.transform(H1B_visa['SECONDARY_ENTITY_1'])
H1B_visa['SECONDARY_ENTITY_1_N'].value_counts()
```

```
0    314508
1    148400
Name: SECONDARY_ENTITY_1_N, dtype: int64
```

```
le = preprocessing.LabelEncoder()
le.fit(H1B_visa.CONTINUED_EMPLOYMENT)
# print list(le.classes_)
H1B_visa['CONTINUED_EMPLOYMENT_N']=le.transform(H1B_visa['CONTINUED_EMPLOYMENT'])
H1B_visa['CONTINUED_EMPLOYMENT_N'].value_counts()
```

```
0    453848
5      3554
2      2262
4      1213
1       944
3       618
6       469
Name: CONTINUED_EMPLOYMENT_N, dtype: int64
```

SUPPORT_H1B, WILLFUL_VIOLATOR, H-1B_DEPENDENT, CONTINUED_EMPLOYMENT, AGENT_REPRESENTING_EMPLOYER, TOTAL_WORKER_POSITIONS are also converted using Label Encoding technique.

The list of dropping down list is depicted here.

```
le = preprocessing.LabelEncoder()
le.fit(H1B_visa['H-1B_DEPENDENT'])
# print list(le.classes_)
H1B_visa['H-1B_DEPENDENT_N']=le.transform(H1B_visa['H-1B_DEPENDENT'])
H1B_visa['H-1B_DEPENDENT_N'].value_counts()
```

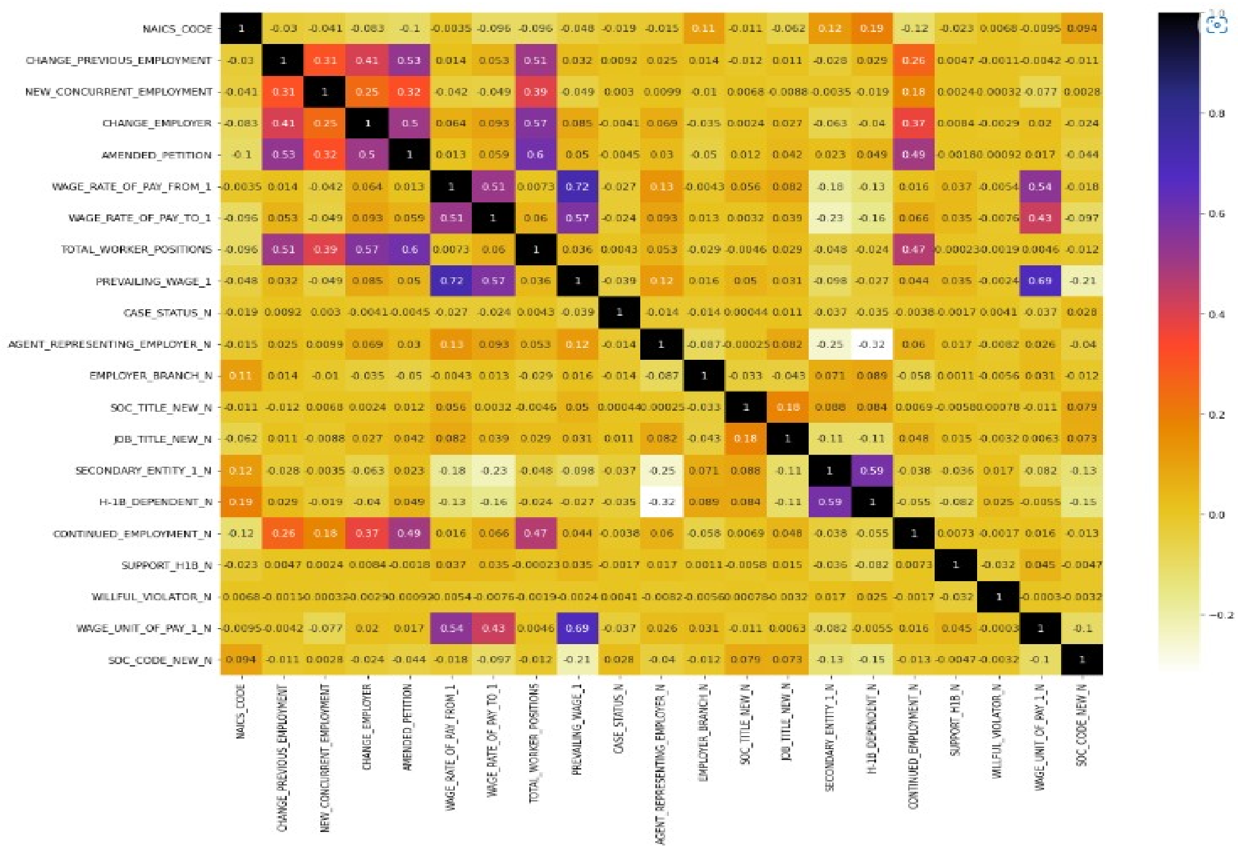
```
0    318930
1    143978
Name: H-1B_DEPENDENT_N, dtype: int64
```

```
H1B_visa.drop('CASE_STATUS', axis=1, inplace=True)
H1B_visa.drop('AGENT_REPRESENTING_EMPLOYER', axis=1, inplace=True)
H1B_visa.drop('EMPLOYER_BRANCH', axis=1, inplace=True)
#H1B_visa.drop('EMPLOYER_NAME', axis=1, inplace=True)
H1B_visa.drop('SOC_CODE_NEW', axis=1, inplace=True)
H1B_visa.drop('JOB_TITLE_NEW', axis=1, inplace=True)
#H1B_visa.drop('SOC_TITLE', axis=1, inplace=True)
#H1B_visa.drop('JOB_TITLE', axis=1, inplace=True)
H1B_visa.drop('SECONDARY_ENTITY_1', axis=1, inplace=True)
H1B_visa.drop('CONTINUED_EMPLOYMENT', axis=1, inplace=True)
H1B_visa.drop('H-1B_DEPENDENT', axis=1, inplace=True)
H1B_visa.drop('SUPPORT_H1B', axis=1, inplace=True)
H1B_visa.drop('WILLFUL_VIOLATOR', axis=1, inplace=True)
H1B_visa.drop('WAGE_UNIT_OF_PAY_1', axis=1, inplace=True)
H1B_visa.drop('SOC_TITLE_NEW', axis=1, inplace=True)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 462908 entries, 24 to 664615
Data columns (total 33 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   CASE_STATUS                    462908 non-null  object
 1   AGENT_REPRESENTING_EMPLOYER    462908 non-null  object
 2   SECONDARY_ENTITY_1             462908 non-null  object
 3   NAICS_CODE                     462908 non-null  float64
 4   CONTINUED_EMPLOYMENT           462908 non-null  object
 5   CHANGE_PREVIOUS_EMPLOYMENT     462908 non-null  int64
 6   NEW_CONCURRENT_EMPLOYMENT      462908 non-null  float64
 7   CHANGE_EMPLOYER                462908 non-null  int64
 8   AMENDED_PETITION               462908 non-null  int64
 9   H-1B_DEPENDENT                 462908 non-null  object
 10  SUPPORT_H1B                    462908 non-null  object
 11  WILLFUL_VIOLATOR               462908 non-null  object
 12  WAGE_RATE_OF_PAY_FROM_1        462908 non-null  float64
 13  WAGE_RATE_OF_PAY_TO_1          462908 non-null  float64
 14  WAGE_UNIT_OF_PAY_1             462908 non-null  object
 15  TOTAL_WORKER_POSITIONS         462908 non-null  float64
 16  PREVAILING_WAGE_1              462908 non-null  float64
 17  EMPLOYER_BRANCH                462908 non-null  object
 18  SOC_TITLE_NEW                  462908 non-null  object
 19  JOB_TITLE_NEW                  462908 non-null  object
 20  SOC_CODE_NEW                   462908 non-null  object
 21  CASE_STATUS_N                  462908 non-null  int32
 22  AGENT_REPRESENTING_EMPLOYER_N  462908 non-null  int32
 23  EMPLOYER_BRANCH_N              462908 non-null  int32
 24  SOC_TITLE_NEW_N                462908 non-null  int32
 25  JOB_TITLE_NEW_N                462908 non-null  int32
 26  SECONDARY_ENTITY_1_N           462908 non-null  int32
 27  H-1B_DEPENDENT_N               462908 non-null  int32
 28  CONTINUED_EMPLOYMENT_N         462908 non-null  int32
 29  SUPPORT_H1B_N                  462908 non-null  int32
 30  WILLFUL_VIOLATOR_N             462908 non-null  int32
 31  WAGE_UNIT_OF_PAY_1_N           462908 non-null  int32
 32  SOC_CODE_NEW_N                 462908 non-null  int32
dtypes: float64(6), int32(12), int64(3), object(12)
memory usage: 98.9+ MB
```

```
Int64Index: 462908 entries, 24 to 664615
Data columns (total 21 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   NAICS_CODE                     462908 non-null  float64
 1   CHANGE_PREVIOUS_EMPLOYMENT     462908 non-null  int64
 2   NEW_CONCURRENT_EMPLOYMENT      462908 non-null  float64
 3   CHANGE_EMPLOYER                462908 non-null  int64
 4   AMENDED_PETITION               462908 non-null  int64
 5   WAGE_RATE_OF_PAY_FROM_1        462908 non-null  float64
 6   WAGE_RATE_OF_PAY_TO_1          462908 non-null  float64
 7   TOTAL_WORKER_POSITIONS         462908 non-null  float64
 8   PREVAILING_WAGE_1              462908 non-null  float64
 9   CASE_STATUS_N                  462908 non-null  int32
 10  AGENT_REPRESENTING_EMPLOYER_N  462908 non-null  int32
 11  EMPLOYER_BRANCH_N              462908 non-null  int32
 12  SOC_TITLE_NEW_N                462908 non-null  int32
 13  JOB_TITLE_NEW_N                462908 non-null  int32
 14  SECONDARY_ENTITY_1_N           462908 non-null  int32
 15  H-1B_DEPENDENT_N               462908 non-null  int32
 16  CONTINUED_EMPLOYMENT_N         462908 non-null  int32
 17  SUPPORT_H1B_N                  462908 non-null  int32
 18  WILLFUL_VIOLATOR_N             462908 non-null  int32
 19  WAGE_UNIT_OF_PAY_1_N           462908 non-null  int32
 20  SOC_CODE_NEW_N                 462908 non-null  int32
dtypes: float64(6), int32(12), int64(3)
memory usage: 56.5 MB
```

Once all the features is numerical converted and created new features with modified name, now still there are old features which has string and object data types with same name. In the next step, dropping down the object data types  then all features can be correlated and plotted in the following figures.



In the above figure, It is shown that PREVAILING_WAGE_1 is highly correlated with WAGE_RATE_OF_PAY_FROM_1 which is 0.72, WAGE_UNIT_OF_PAY_1_N is correlated with PREVAILING_WAGE_1 is 0.69.

But Input features are mutually dependent on each other. Here target features is CASE_STATUS_N is highly correlated with SOC_CODE_N has 0.028. Even though this considered features does not effective of ouput feature CASE_STATUS_N. The Follow up modelling part will be implemented.

For the modelling, the focus is moved into two types of features. One is considered as employee skillset based H1-B Visa approval and other is looked for how wage rate information is affecting the H1-B Visa. Further considerations of featuring set and approaches will be discussed in the Modelling part.

## 4. ML Modelling

In this section, The brief description of proposed ML modelling techniques is discussed along with respective results is produced. In the ML modelling part, Navies Bayes classifier and Decision Tree classifier is used with Employee skill set and Employee wage rate information.

The features of Employee skill set consists of EMPLOYEE_BRANCH, JOB_TITLE_N, SOC_CODE_N, SOC_TITLE_N, and NAICS_CODE_N.

The feature set of wage rate information of Employee has PREVAILING_WAGE_1, WAGE_RATE_OF_PAY_FROM_1, WAGE_RATE_OF_PAY_TO_1, and

WAGE_UNIT_OF_PAY_1_N.

Here, Four cases will be implemented in these subsection.

### 4a. Navies Bayes Classifier:

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

1. Gaussian: It is used in classification and it assumes that features follow a normal distribution.

2. Multinomial: It is used for discrete counts. For example, let's say, we have a text classification problem. Here we can consider Bernoulli trials which is one step further and instead of "word occurring in the document", we have "count how often word occurs in the document", you can think of it as "number of times outcome number x_i is observed over the n trials".

3. Bernoulli: The binomial model is useful if your feature vectors are binary (i.e., zeros and ones). One application would be text classification with 'bag of words' model where the 1s & 0s are "word occurs in the document" and "word does not occur in the document" respectively.

### i) Multinomial Navies Bayes algorithm for Features of Employee skill set:

The first algorithm we used is the Multinomial Naive Bayes to build our model.

```
select_columns_for_MNB = ['JOB_TITLE_NEW_N','EMPLOYER_BRANCH_N','SOC_CODE_NEW_N','SOC_TITLE_NEW_N','NAICS_CODE']
H1B_visa_MNB = H1B_visa[select_columns_for_MNB]
```

```
from sklearn.model_selection import train_test_split
x_train,x_val,y_train,y_val=train_test_split(X,y,test_size=0.4,random_state=42)
```

```
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB()
clf.fit(x_train, y_train)
```

```
MultinomialNB()
```

```
from sklearn import metrics
metrics.accuracy_score(y_val, predictions)
```

```
0.9883616685748849
```

Since the data was very imbalanced, we used the concept of SMOTE to oversample the data. It involves randomly selecting examples from the minority class, with replacements, and adding them to the training dataset.

Imbalanced dataset: In this dataset, CASE_STATUS is the target value and on doing EDA we find that the labelled values has an uneven distribution of features. Hence to save the model from getting over fit, used the method SMOTE.

SMOTE: This technique involves creating a new dataset by oversampling observations from the minority class, which produces a dataset that has more balanced classes.

After that, we performed the training of data and formed the confusion matrix and accuracy score for our model. The accuracy of our model turned out to be ____91__ %

Metrics & Results:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.99 | 0.92 | 0.95 | 88698 |
| 1 | 0.02 | 0.18 | 0.04 | 824 |
| accuracy | | | 0.91 | 89522 |
| macro avg | 0.51 | 0.55 | 0.50 | 89522 |
| weighted avg | 0.98 | 0.91 | 0.95 | 89522 |

**ii) Multinomial Navies Bayes algorithm for Features of Employee wage information:**

Implementation of wage rate information is depicted in the bottom of the line and the results will be discussed.

```
metrics.accuracy_score(y_val1, predictions)
```

```
0.6363513630276062
```

```
print(confusion_matrix(y_val1, predictions))
print(classification_report(y_val1, predictions))
```

```
[[116998  66256]
 [  1083    839]]
              precision    recall  f1-score   support

           0       0.99      0.64      0.78    183254
           1       0.01      0.44      0.02      1922

    accuracy                           0.64    185176
   macro avg       0.50      0.54      0.40    185176
weighted avg       0.98      0.64      0.77    185176
```

```
select_column_wage_rate = ['WAGE_RATE_OF_PAY_FROM_1', 'WAGE_RATE_OF_PAY_TO_1','WAGE_UNIT_OF_PAY_1_N','PREVAILING_WAGE_1']
H1B_visa_wage = H1B_visa[select_column_wage_rate]
```

From the above, the same results has been observed then later applied oversampling with SMOTE. But, it has very low accuracy after oversampling SMOTE.

### 4b. Decision Tree Classifier:

### i) Decision Tree algorithm for Features of Employee skill set:

The second algorithm we used is the Decision Tree. It is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test=train_test_split(X,y,test_size=0.4,random_state=42)
```

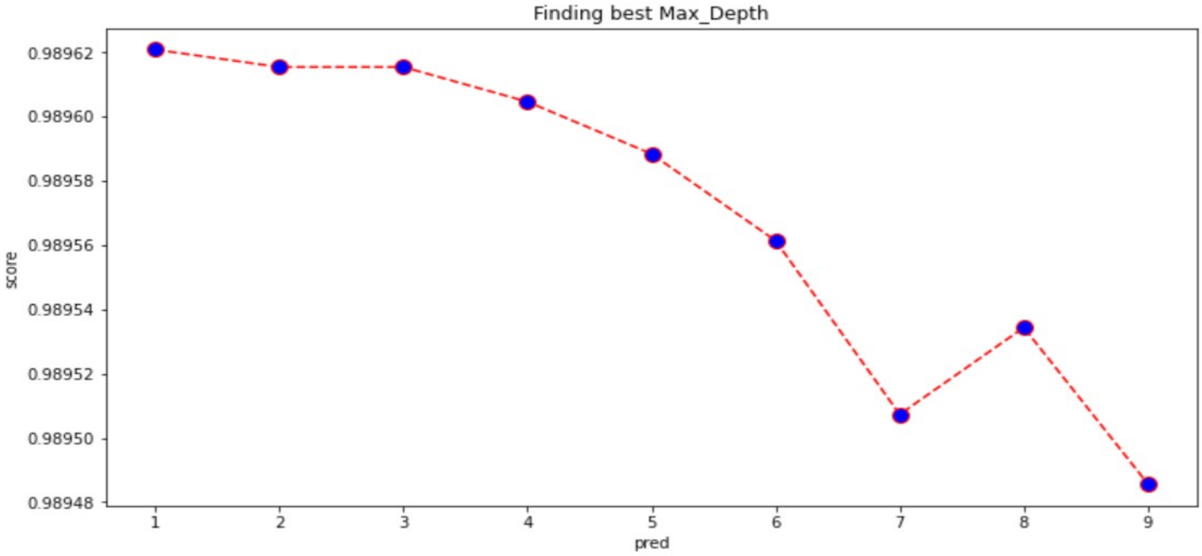
```
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
```

Similar to Multinomial Naive Bayes, two feature sets will be used, one that contains details on occupation and company name, and the other on the wage-related details. The model is performed by the SMOTE here as well since the data is very imbalanced.

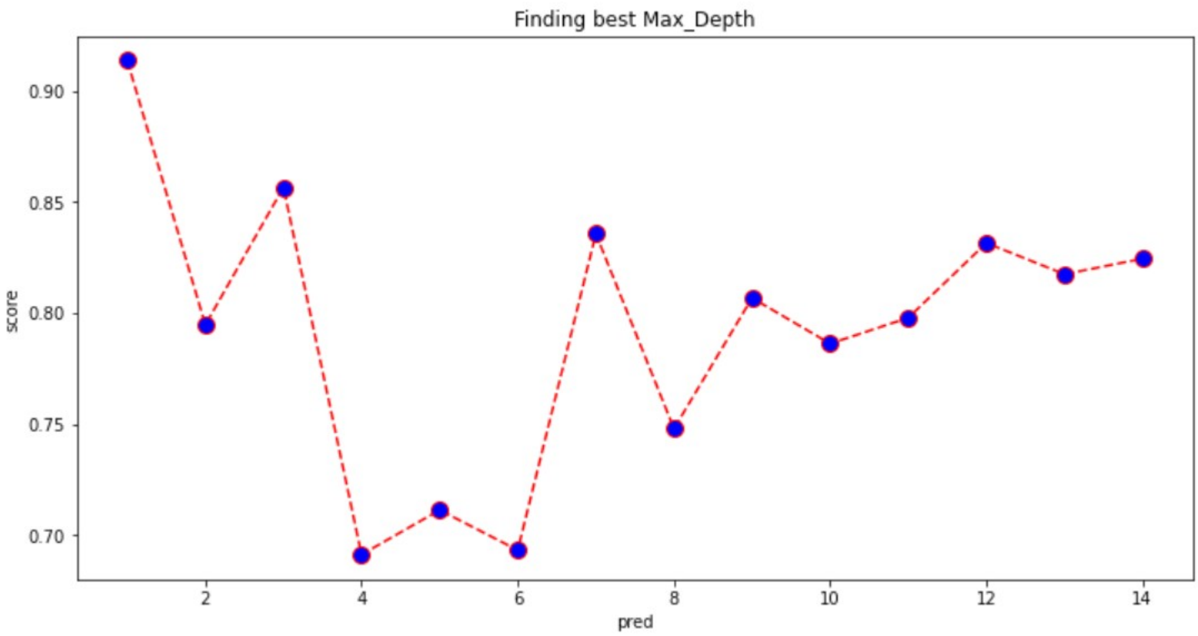
After doing the necessary EDA and Data Preprocessing, split our model into train and test sets:

x_train, x_val, y_train, y_val =train_test_split(X, y, test_size=0.4, random_state=42)

It is obtained to perform the training using Decision Tree Classifier with criterion as entropy with a maximum dept as 4: In the following figure, shows the maximum depth before Imbalanced dataset.

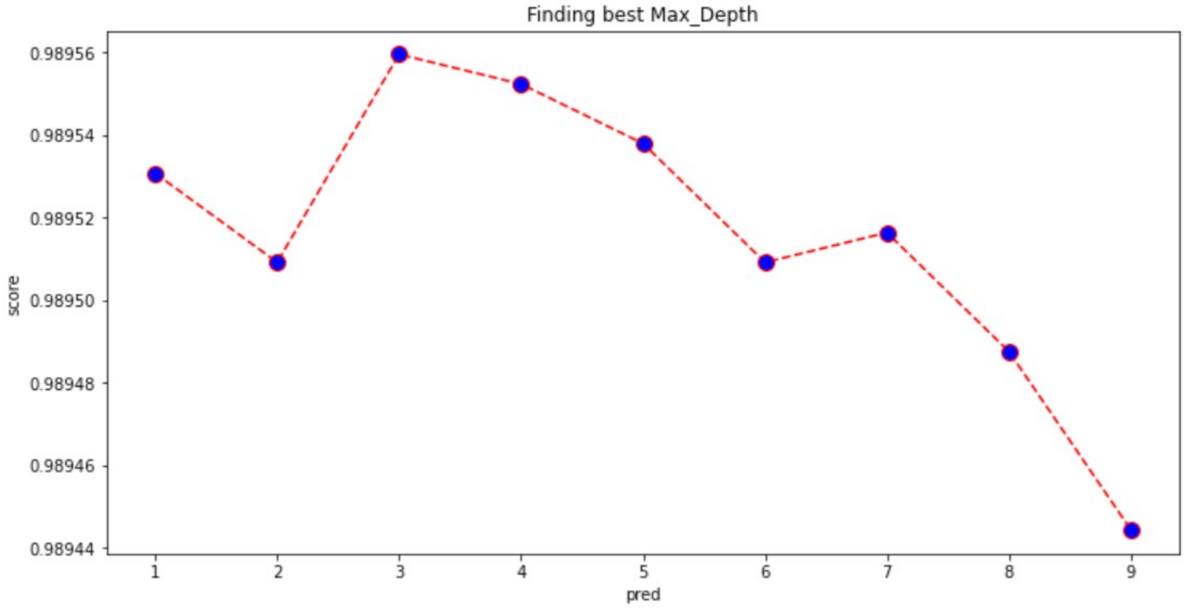



It is obtained to perform the training using Decision Tree Classifier with criterion as entropy with a maximum dept as 4: In the following figure, shows the maximum depth after Imbalanced dataset.

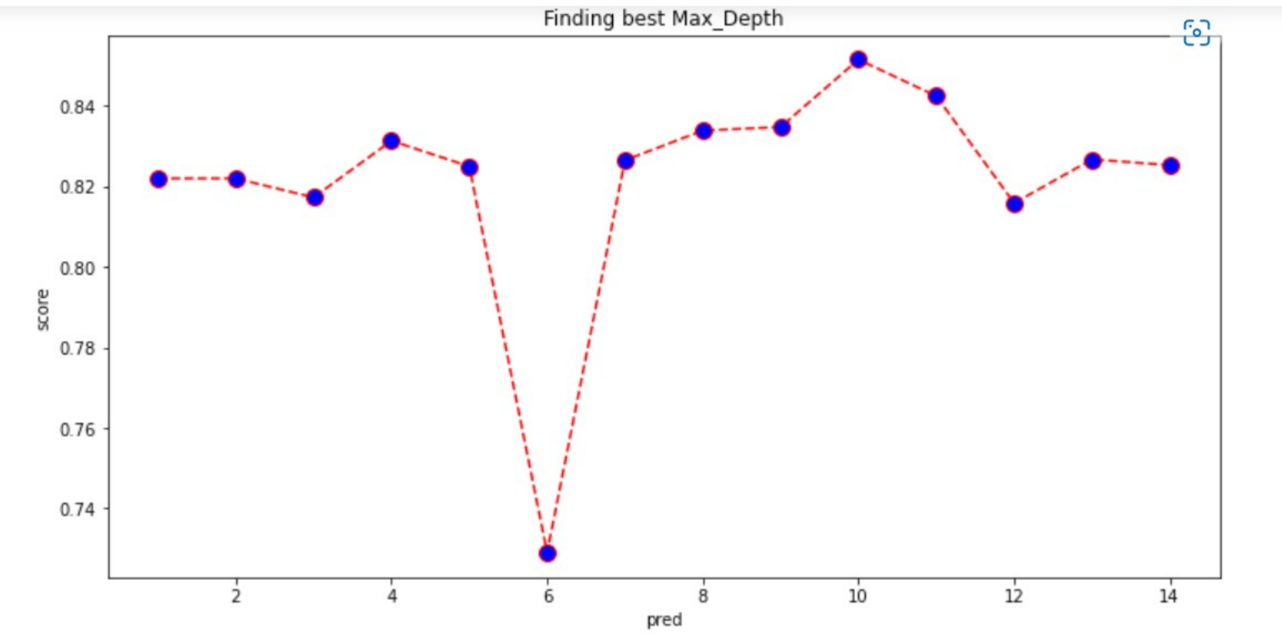



**ii) Decision Tree algorithm for Features of Employee wage information:**

Wage information is calculated and structure in unique buckets based on its distribution from features WAGE_RATE_OF_PAY_FROM_N, WAGE_RATE_OF_PAY_TO_N, WAGE_UNIT_OF_PAY_N, and PREVAILING_WAGE_1.




The model is performed the training using Decision Tree Classifier with criterion as entropy with a maximum depth as 4: After SMOTE, the model is performed with the training of data and formed the confusion matrix and accuracy score. The accuracy of the model turned out to be ____85__ %

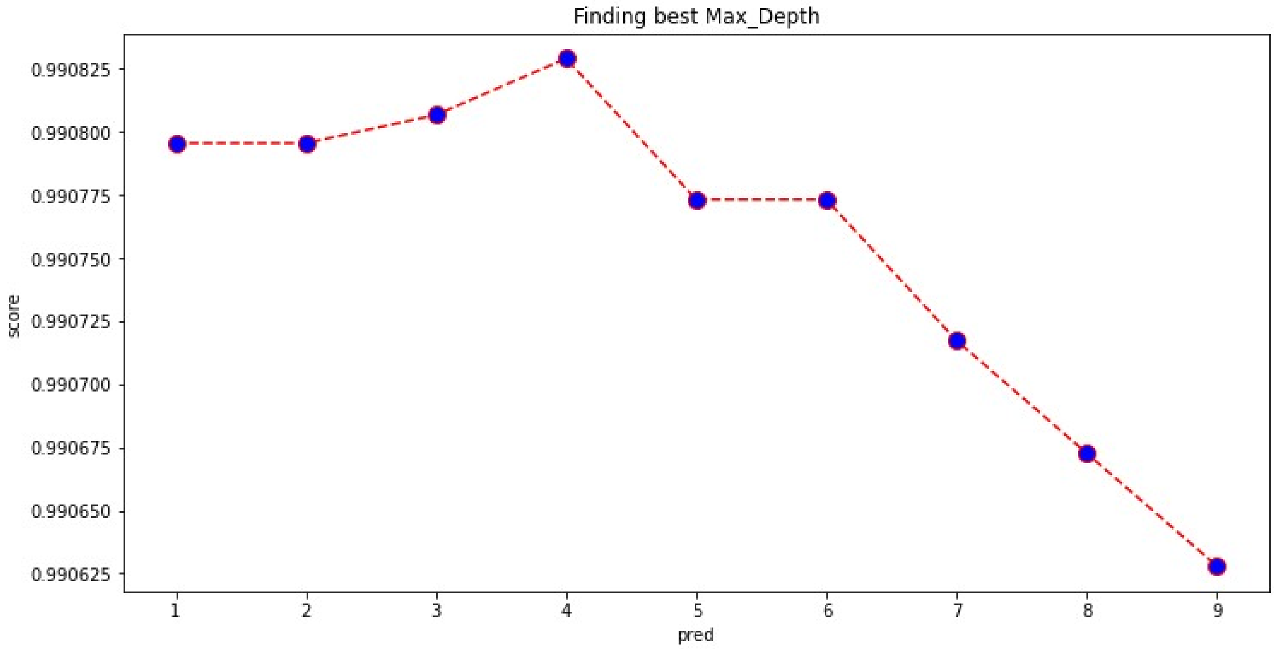


## 4c. Additional feature set

So far, the Four cases have been performed and other additional model is the combination Feature of Employee skill set and Feature of Wage rate information.

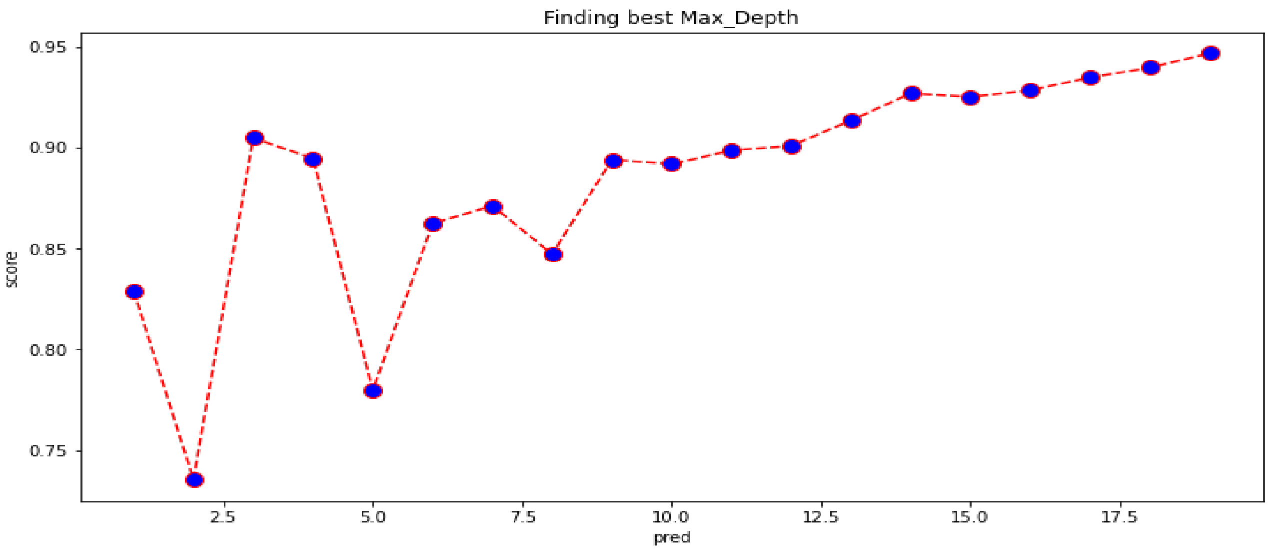
```
x_features = dataset[['SOC_TITLE_NEW','EMPLOYER_BRANCH','JOB_TITLE_NEW','SOC_CODE_NEW','WAGE_UNIT_OF_PAY_1','WAGE_RATE_OF_PAY_TO_1'
,'WAGE_RATE_OF_PAY_FROM_1','PREVAILING_WAGE_1']]
```

Decision Tree is used for the additional feature set, it has accuracy score before resampling is 99.07%. Max depth of the figure is plotted here.




After SMOTE, the accuracy score is much quite higher than all remaining models.

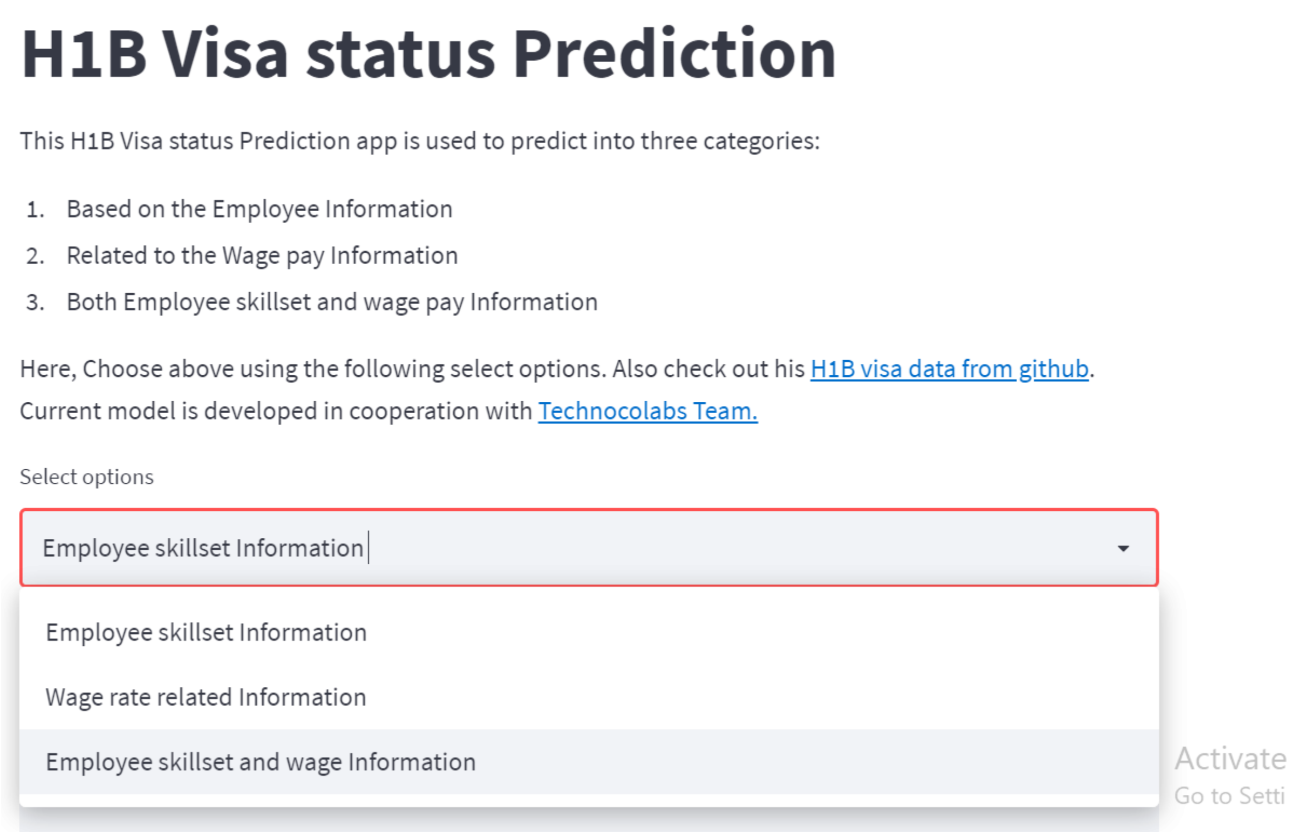
Accuracy score after resampling is 94%.



## 5. Deployment

In this section, deployment of the best three models are created the multi-web page using streamlit library of python and Heroku app.

Out of 5 cases of models are developed in the modelling part, only three models are deployed as final stage in user interface. The following three models have the best accuracy score after resampling. They are:




1. Multinomial Navies Bayes Classifier for Employee skill set features.

2. Decision Tree classifier for Wage rate information features.

3. Decision Tree classifier for both skill set and wage rate information.

In the next page, the first look of the Multiweb page app is presented and find the link of the web page after the figure.

https://h1b-visa-prediction.herokuapp.com

## 6.Conclusion:

- Through data visualization, proposed features are interpreted and evaluated.
- Using bin counting, Categorical attributes reduce the large number categorical variables.
- Analysed the target feature with independent features using correlation graphs.
- After modelling part, decision tree classifier prediction and accuracy higher than multinomial navies bayes classifier
- wage related information data set features has higher accuracy with decision tree classifier.
- Set of employee skill set and set of wage rate information is combined it has very good accuracy 94% using decision tree classifier.
- Developed the multiweb page using Heroku app and streamlit library.

Citations:

1.https://blog.streamlit.io/introducing-multipage-apps/

2. https://github.com/mlp9/Comprehensive-H1-B-Visa-Data-Analysis-using-Python/blob/main/H1-B%20visa%20analysis.ipynb

3. Feature Engineering: Bayesian Methods for Binning | by Andy Greatorex | Towards Data Science

4. sklearn.naive_bayes.MultinomialNB — scikit-learn 1.1.1 documentation

5. sklearn.tree.DecisionTreeClassifier — scikit-learn 1.1.1 documentation

List of Team B:

Sai Syamsunder Reddy Nallamilli(Lead)

Abhinav Arun

Joel Kennedy

Srivaikunthan