PROJECT REPORT

Predicting US H-1B Visa Approvals

SYST-568

Applied Predictive Analytics

Instructor: Dr. Ran Ji

Team:

Shravya Gaddam (G01209938)

Chandana Narla (G01211387)

Meghana Gunuganti (G01194548)

Anirudh Tunuguntla (G01208856)

Una Suman Kumar Patro (G01224054)

Abstract:

This project deals with H-1B Visa Predictions data taken from the Kaggle Dataset for the year 2018. This dataset includes over 6 lakh records of applicants who have applied for the non-immigrant work permit H-1B Visa status for different employer countries with different prevailing factors. This project deals with the H-1B visa type for the USA as the employer country. H-1B visa basically deals with the work authorization granted by the country to any non-immigrant. To attain this Visa status, the applicant's profile must meet certain criteria like wages, job role, work location etc. Hence, based on the previous data, we built a few predictive models, which evaluates the prime factors for the Visa acceptance and predicts the response for any new applicant.

Objective:

In this project, we intend to forecast the possible result of H1-B visa applications which are submitted by their employers for specialty jobs for several highly qualified foreign citizens in the United States. Every year, the employers file more than a million visa applications and only 65,000 petitions are accepted.

The goal, therefore, is to analyze the petitions filed and their results for the 2018 year and to find a trend for interpreting the results using a predictive model developed using classification techniques

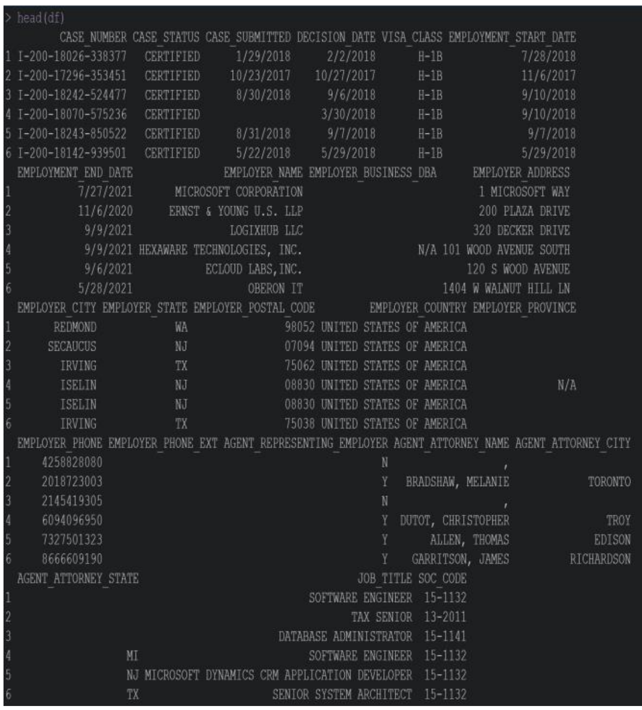
Introduction:

The H1B Visa is a widely sought-after non-immigrant visa that requires foreign professionals to enter the country in their respective specialty professions. H-1B visas are a subset of non-immigrant, income-based visas for short term foreign employees in the US. A US employer must give them a position and send a petition for an H-1B visa to the U.S. immigration department for a foreign citizen to apply for an H1-B visa.

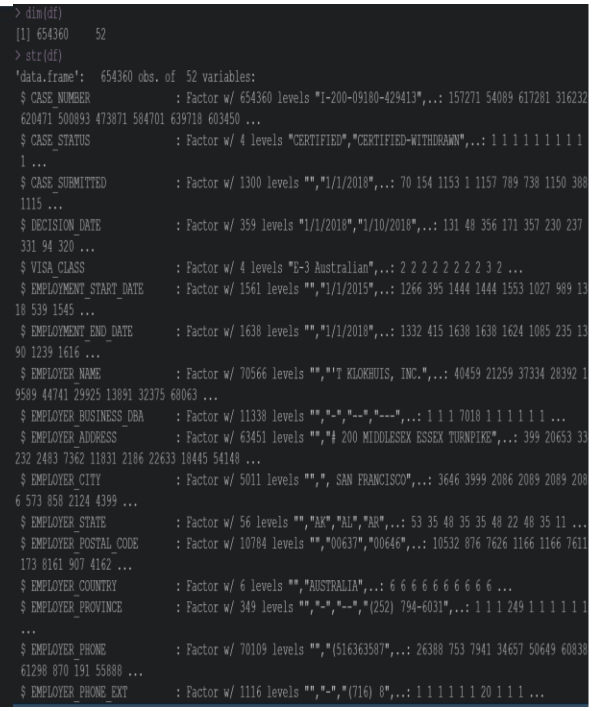
In this project, we intend to forecast the possible result of H1-B visa applications which are submitted by their employers for specialty jobs for several highly qualified foreign citizens in the United States. Every year, the employers file more than a million visa applications and only 65,000 petitions are accepted. The goal, therefore, is to analyze the petitions filed and their results for the 2018 year and to find a trend for interpreting the results using a predictive model developed using regression techniques.

Dataset Description:

This dataset is taken from Kaggle source where the dataset was prepared by Abishek Anbarasan. It consists of 654361 rows and 52 columns which includes the data related to the total number of Visa Applications submitted for the year 2018. This data includes details about each visa application case which is uniquely identified by a Case Number assigned to each application. In order to understand the decision and duration taken for a decision is specified using Case Submission Date and Case Decision Dates. The status of each application is classified as Certified, Withdrawn and Denied based on factors like Visa Type for which they applied, Employer Country, Wages, Duration of the employment. This case status shall be converted into the binary response variable for further modelling purposes. Some other factors also include whether a particular applicant used an agent in order to file the H1B case. Also, there exist records about whether the applicant has previously applied for any kind of Visa types. The information about an applicant being a Full-Time employee or working on Contract basis is provided. For every job type there exists a SOC_Code and SOC_Name which is Standard Occupation Classification, authorized and approved by the Government which explicitly classifies the Job Role and Job Specifications which also play a major role while picking for visas. The wages received by an applicant and the unit of pay details like Annual, biweekly or hourly pay detail also add up to be a significant factor while considering for H-1B status. Few H1B visas are filed as dependents, hence this information also adds up to be a factor for a specific visa type to get picked. Other generic information about the Employer company, its location and contact details, Agent contact details and work location related details. Clearly, as per the objective of the project, the newly modified Case Status column will be the dependent variable for which the models are trained to predict the Status of an Application based on the above-mentioned factors.



```
> head(df)
       CASE_NUMBER CASE_STATUS CASE_SUBMITTED DECISION_DATE VISA_CLASS EMPLOYMENT_START_DATE
1 I-200-18026-338377   CERTIFIED      1/29/2018      2/2/2018       H-1B             7/28/2018
2 I-200-17296-353451   CERTIFIED     10/23/2017    10/27/2017       H-1B             11/6/2017
3 I-200-18242-524477   CERTIFIED      8/30/2018      9/6/2018       H-1B             9/10/2018
4 I-200-18070-575236   CERTIFIED                    3/30/2018       H-1B             9/10/2018
5 I-200-18243-850522   CERTIFIED      8/31/2018      9/7/2018       H-1B              9/7/2018
6 I-200-18142-939501   CERTIFIED      5/22/2018     5/29/2018       H-1B             5/29/2018
  EMPLOYMENT_END_DATE              EMPLOYER_NAME EMPLOYER_BUSINESS_DBA      EMPLOYER_ADDRESS
1           7/27/2021      MICROSOFT CORPORATION                              1 MICROSOFT WAY
2           11/6/2020      ERNST & YOUNG U.S. LLP                             200 PLAZA DRIVE
3            9/9/2021                LOGIXHUB LLC                            320 DECKER DRIVE
4            9/9/2021 HEXAWARE TECHNOLOGIES, INC.                 N/A 101 WOOD AVENUE SOUTH
5            9/6/2021             ECLOUD LABS,INC.                         120 S WOOD AVENUE
6           5/28/2021                   OBERON IT                       1404 W WALNUT HILL LN
  EMPLOYER_CITY EMPLOYER_STATE EMPLOYER_POSTAL_CODE         EMPLOYER_COUNTRY EMPLOYER_PROVINCE
1      REDMOND             WA                98052 UNITED STATES OF AMERICA
2     SECAUCUS             NJ                07094 UNITED STATES OF AMERICA
3       IRVING             TX                75062 UNITED STATES OF AMERICA
4       ISELIN             NJ                08830 UNITED STATES OF AMERICA               N/A
5       ISELIN             NJ                08830 UNITED STATES OF AMERICA
6       IRVING             TX                75038 UNITED STATES OF AMERICA
  EMPLOYER_PHONE EMPLOYER_PHONE_EXT AGENT_REPRESENTING_EMPLOYER AGENT_ATTORNEY_NAME AGENT_ATTORNEY_CITY
1     4258828080                                              N                   ,
2     2018723003                                              Y   BRADSHAW, MELANIE             TORONTO
3     2145419305                                              N                   ,
4     6094096950                                              Y  DUTOT, CHRISTOPHER                TROY
5     7327501323                                              Y       ALLEN, THOMAS              EDISON
6     8666609190                                              Y    GARRITSON, JAMES          RICHARDSON
  AGENT_ATTORNEY_STATE                                  JOB_TITLE SOC_CODE
1                                             SOFTWARE ENGINEER  15-1132
2                                                    TAX SENIOR  13-2011
3                                        DATABASE ADMINISTRATOR  15-1141
4                   MI                        SOFTWARE ENGINEER  15-1132
5                   NJ MICROSOFT DYNAMICS CRM APPLICATION DEVELOPER  15-1132
6                   TX                  SENIOR SYSTEM ARCHITECT  15-1132
```

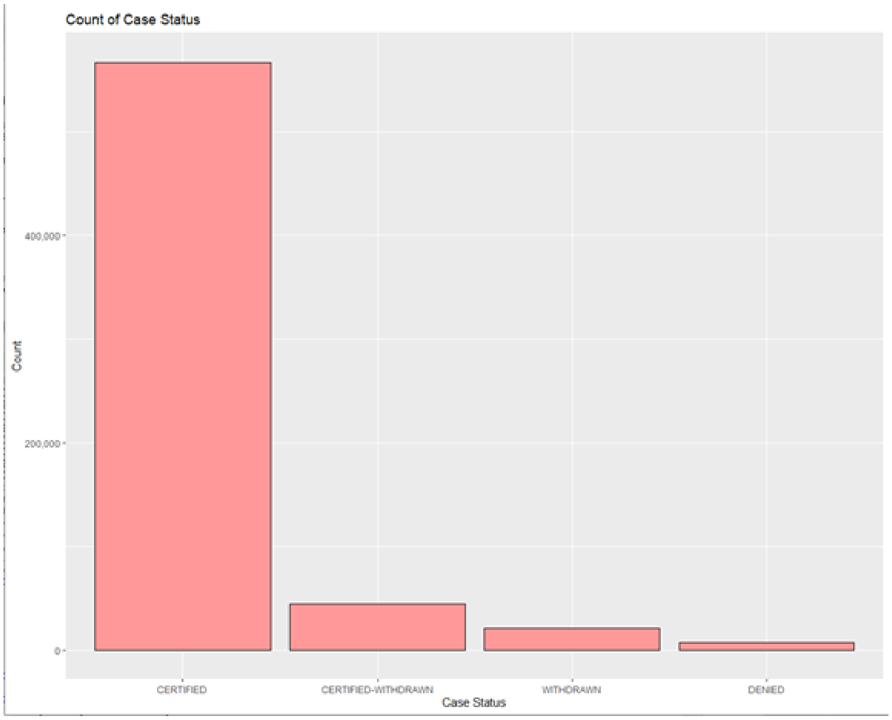


```
> dim(df)
[1] 654360     52
> str(df)
'data.frame':   654360 obs. of  52 variables:
 $ CASE_NUMBER              : Factor w/ 654360 levels "I-200-09180-429413",..: 157271 54089 617281 316232
 620471 500893 473871 584701 639718 603450 ...
 $ CASE_STATUS              : Factor w/ 4 levels "CERTIFIED","CERTIFIED-WITHDRAWN",..: 1 1 1 1 1 1 1 1 1
 1 ...
 $ CASE_SUBMITTED           : Factor w/ 1300 levels "","1/1/2018",..: 70 154 1153 1 1157 789 738 1150 388
 1115 ...
 $ DECISION_DATE            : Factor w/ 359 levels "1/1/2018","1/10/2018",..: 131 48 356 171 357 230 237
 331 94 320 ...
 $ VISA_CLASS               : Factor w/ 4 levels "E-3 Australian",..: 2 2 2 2 2 2 2 2 3 2 ...
 $ EMPLOYMENT_START_DATE    : Factor w/ 1561 levels "","1/1/2015",..: 1266 395 1444 1444 1553 1027 989 13
18 539 1545 ...
 $ EMPLOYMENT_END_DATE      : Factor w/ 1638 levels "","1/1/2018",..: 1332 415 1638 1638 1624 1085 235 13
90 1239 1616 ...
 $ EMPLOYER_NAME            : Factor w/ 70566 levels "","'T KLOKHUIS, INC.",..: 40459 21259 37334 28392 1
9589 44741 29925 13891 32375 68063 ...
 $ EMPLOYER_BUSINESS_DBA    : Factor w/ 11338 levels "","-","--","---",..: 1 1 1 7018 1 1 1 1 1 1 ...
 $ EMPLOYER_ADDRESS         : Factor w/ 63451 levels "","# 200 MIDDLESEX ESSEX TURNPIKE",..: 399 20653 33
232 2483 7362 11831 2186 22633 18445 54148 ...
 $ EMPLOYER_CITY            : Factor w/ 5011 levels "",", SAN FRANCISCO",..: 3646 3999 2086 2089 2089 208
6 573 858 2124 4399 ...
 $ EMPLOYER_STATE           : Factor w/ 56 levels "","AK","AL","AR",..: 53 35 48 35 35 48 22 48 35 11 ...
 $ EMPLOYER_POSTAL_CODE     : Factor w/ 10784 levels "","00637","00646",..: 10532 876 7626 1166 1166 7611
 173 8161 907 4162 ...
 $ EMPLOYER_COUNTRY         : Factor w/ 6 levels "","AUSTRALIA",..: 6 6 6 6 6 6 6 6 6 6 ...
 $ EMPLOYER_PROVINCE        : Factor w/ 349 levels "","-","--","(252) 794-6031",..: 1 1 1 249 1 1 1 1 1 1
 ...
 $ EMPLOYER_PHONE           : Factor w/ 70109 levels "","(516363587",..: 26388 753 7941 34657 50649 60838
 61298 870 191 55888 ...
 $ EMPLOYER_PHONE_EXT       : Factor w/ 1116 levels "","-","(716) 8",..: 1 1 1 1 1 1 20 1 1 1 ...
```

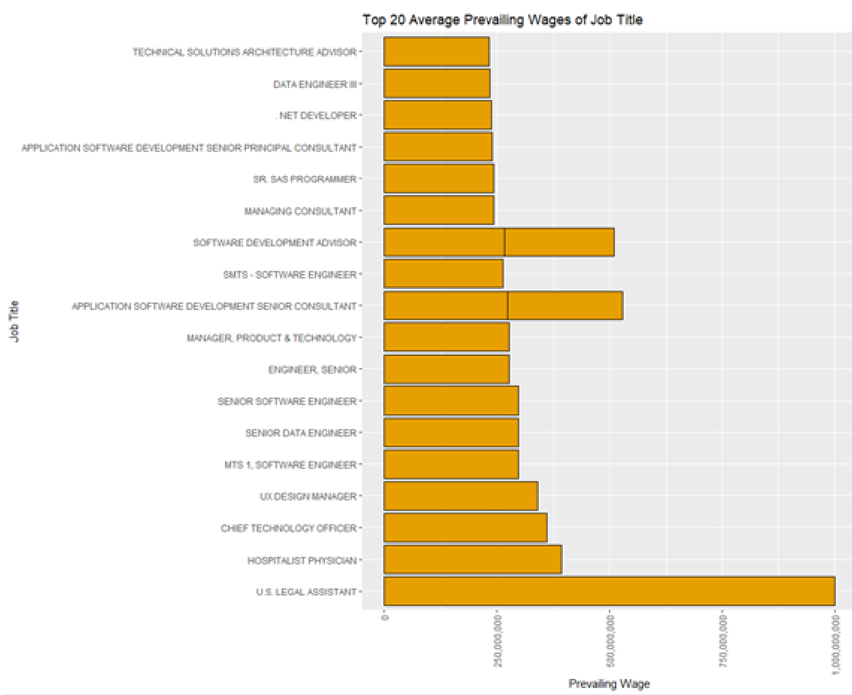
(Anbarasan, 2019)

Exploratory Data Analysis and Visualizations:

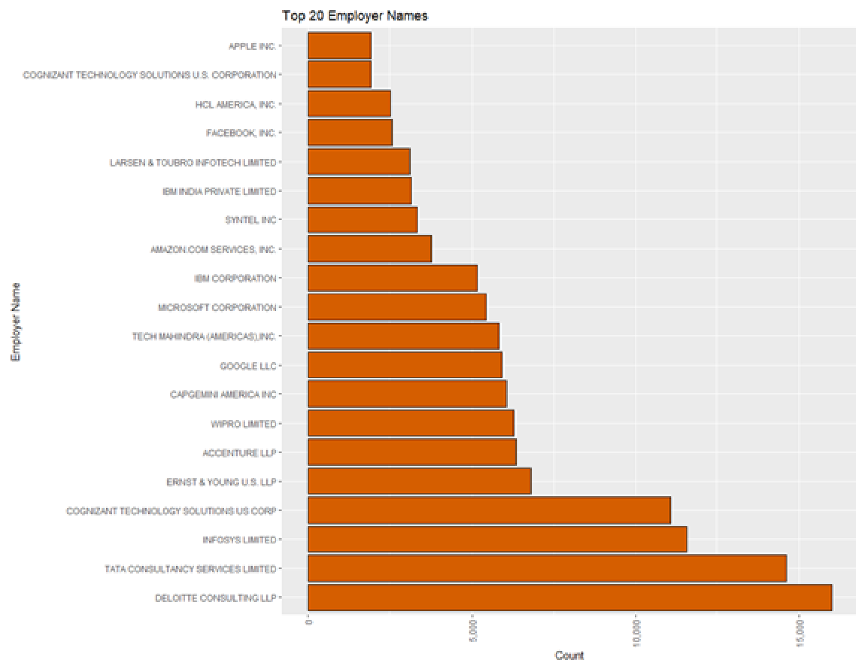
Analyzing the significance of each column, the value levels and other factors can be done through visualizations. It is difficult to understand the range of value just by looking at the dataset. As the objective of the project, we wanted to predict whether a particular applicant will be Certified or not with the Visa. Hence from the below figure, we could understand that most important categories are **certified** and **denied.** Here the certified are stated as approvals which are about half a million which is over 3/4th of the total applications and the number of denied cases are around 10k. Also, there are two other categories namely Certified Withdrawn and Withdrawn where Certified Withdrawn denotes that their case has been approved but they have Withdrawn the file and Withdrawn is nothing but withdrawn.



The below visualization shows the Top 20 job titles based on their average wages. Since we know that about 3/4th applicants have been approved with the visa, we wanted to know the other prevailing significant factors. Here the US legal assistant company has the highest salary average with million dollars per year. Later different Software employee roles and Hospital Physicians receive higher wages.



While understanding the Visa class, the employer who is employing the applicant plays a major role. Hence the below graph shows the top 20 company (employer) names with the count of the number of applicants applying from their company. Deloitte has applied for Visa for over 15000 employees and followed by TCS for 14000 employees.



By analyzing the box plot, between Certified and Denied case status, based on the wage distribution of the applicants. We could see, although the box plot of certified cases indicates that they have better average wage rate than the denied cases, we can see that the outliers for the denied cases are more which is over 1.5M dollars per annum. So, this clearly indicates that the wage might not be a huge factor for predicting H1 B visa approval

(Jhanji, 2018)

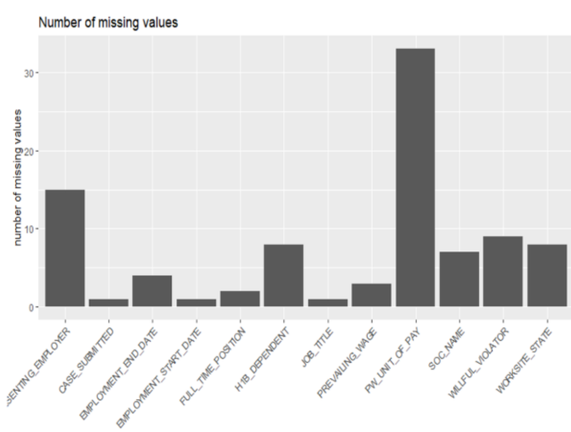
Data Pre-processing:

Data preprocessing is an essential stage before modelling is performed on the data. After EDA and visualizations, certain correlations between the variables, few important values are identified. As a part of data pre-processing, we performed the Data Filtration, Handling Missing Data, Handling Near Zero Variance, Feature Engineering, One Hot Encoding and Resampling for the chosen dataset. After the dataset is processed, the set of variables are fed to the predictive models for further prediction of the visa status.

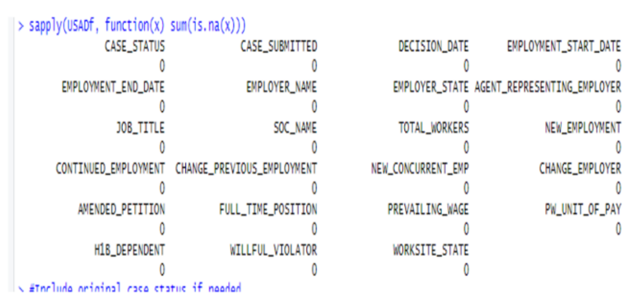
- **Data Filtration:** As per the objective of the project, we wanted to predict the Case status for the Visa type H-1B and employer country as the USA. Hence, as a part of data filtration, we filtered the columns whose Employer country is the USA and Visa Type was H-1B. Later, we removed these columns as the needed information from them is extracted. Also, after understanding the importance and significance of each column, we dropped a few columns which contain genetic information like Postal codes, address, Employee business DBA, agent address and contact information. These columns do not contribute towards acting as predicting variables. Certain uninformative columns have been dropped because of the larger number of null values present in them so that other important information can be retained.

- **Handling Missing Data:** One of the major steps while performing data preprocessing is identifying and handling the missing data. After data filtration and removal of unimportant columns, we checked for NA values (Null values) in the dataset. We could see that not many null values are present. Hence, we omitted the null values since they were not creating a huge difference to the output. Below figure depicts the number of missing values the dataset has before and after omitting the NA values.

Missing values in each Variable



After removing null values



```
> sapply(USADF, function(x) sum(is.na(x)))
           CASE_STATUS          CASE_SUBMITTED            DECISION_DATE    EMPLOYMENT_START_DATE
                     0                       0                        0                        0
   EMPLOYMENT_END_DATE           EMPLOYER_NAME           EMPLOYER_STATE AGENT_REPRESENTING_EMPLOYER
                     0                       0                        0                        0
             JOB_TITLE                SOC_NAME            TOTAL_WORKERS           NEW_EMPLOYMENT
                     0                       0                        0                        0
  CONTINUED_EMPLOYMENT CHANGE_PREVIOUS_EMPLOYMENT     NEW_CONCURRENT_EMP          CHANGE_EMPLOYER
                     0                       0                        0                        0
      AMENDED_PETITION      FULL_TIME_POSITION          PREVAILING_WAGE           PW_UNIT_OF_PAY
                     0                       0                        0                        0
         H1B_DEPENDENT        WILLFUL_VIOLATOR           WORKSITE_STATE
                     0                       0                        0
```

- **Handling Near Zero Variance:** From the concept of Near-zero variance, there are certain columns present in the dataset, which gives a constant value throughout. For example, they only predict the output belonging to one class throughout. Such columns are considered to be outliers and these uninformative columns should be removed in order to improve the predictive power of the model. Handling near-zero variance variables is important as they could lead to misleading and biased outputs of a model. In our dataset, we have found 4 such predictors TOTAL_WORKERS, NEW_CONCURRENT_EMP, FULL_TIME_POSITION, WILLFUL_VIOLATOR. The below shows the near-zero variance columns for the chosen dataset.

```
> #Detecting near zero variance
> x = nearZeroVar(USADf, saveMetrics = TRUE)
> str(x, vec.len=2)
'data.frame':	21 obs. of  4 variables:
 $ freqRatio    : num  1.92 2.64 ...
 $ percentUnique: num  0.00188 0.00188 ...
 $ zeroVar      : logi  FALSE FALSE FALSE ...
 $ nzv          : logi  FALSE FALSE FALSE ...
> x[x[,"zeroVar"] > 0, ]
[1] freqRatio     percentUnique zeroVar       nzv
<0 rows> (or 0-length row.names)
> x[x[,"zeroVar"] + x[,"nzv"] > 0, ]
                    freqRatio percentUnique zeroVar  nzv
TOTAL_WORKERS        30.89685  0.0081325607   FALSE TRUE
NEW_CONCURRENT_EMP  210.01456  0.0014075586   FALSE TRUE
FULL_TIME_POSITION   50.46945  0.0003127908   FALSE TRUE
WILLFUL_VIOLATOR   1978.58204  0.0003127908   FALSE TRUE
> |
```

- **Feature Engineering:** As a part of feature engineering, we have done scaling, refactoring, introduced calculated columns, added new columns, formatted the uncleaned data columns, etc. New Case Status and Employment duration are two new columns which are added to the existing dataset. New case status is the response variable which is of binary type, where the Certified values are treated as 1 and the other statuses are treated as 0. Employment duration is the calculated column, which is calculated using the applicant's employment start and end dates.

  The dates present in the dataset used different conventions in different columns and rows, hence for the better understanding of the model, we converted the date into a single format and picked up only the months for each applicant. This data is fed to the model as one of the predictor variables. Wage levels are present for the different duration for different applicants like yearly, bi-weekly, weekly, hourly etc. Hence all these values have been scaled to one wage rate which is wage per annum using the if-Else functions. Columns like Job_title, Employer name has many levels, hence these columns have to be refactored. We have considered the top 50 levels based on the frequency of the job titles and employer company names. The rest values are categorized as "Others" for better modelling purposes.

- **One Hot Encoding:** It is known that one hot encoding is a process where categorical variables are converted into a form which is given to the machine learning model for better prediction. Here, we chose one hot encoding over the label encoding because in label encoding the assigned numeric value to the label could be treated as order or hierarchy in the data which is not the correct value of the column. Hence in one-hot encoding, the available categorical variables have split into columns based on the labels or unique entities present in the column. These newly formed columns are binary columns with 1 for existence and 0 for nonexistence values. We had 11 categorical variables in our dataset on which we have performed One Hot Encoding which resulted in a total of 306 variables. Below figure represents the newly formed columns after one-hot encoding.

```
> dim(final)
[1] 639405    306
> names(final)
  [1] "TOTAL_WORKERS"
  [2] "NEW_EMPLOYMENT"
  [3] "CONTINUED_EMPLOYMENT"
  [4] "CHANGE_PREVIOUS_EMPLOYMENT"
  [5] "NEW_CONCURRENT_EMP"
  [6] "CHANGE_EMPLOYER"
  [7] "AMENDED_PETITION"
  [8] "PREVAILING_WAGE"
  [9] "EMPLOYMENT_DURATION"
 [10] "CASE_STATUS_NEW"
 [11] "CASE_SUBMITTED.April"
 [12] "CASE_SUBMITTED.August"
 [13] "CASE_SUBMITTED.December"
 [14] "CASE_SUBMITTED.February"
 [15] "CASE_SUBMITTED.January"
 [16] "CASE_SUBMITTED.July"
 [17] "CASE_SUBMITTED.June"
 [18] "CASE_SUBMITTED.March"
 [19] "CASE_SUBMITTED.May"
 [20] "CASE_SUBMITTED.November"
```

- **Resampling:** In Classification of datasets, the algorithms cannot run efficiently if the response variable is imbalanced as they do not get enough cases of the lower class to properly predict. And due to the unequal distribution of classes, the algorithms tend to be biased towards the majority class. So, it is desirable to attain a balanced dataset if not an equal number of classes. So, we have implemented oversampling on our training data to balance the classes. Where the algorithm works on replicating the minority class and balances the data using the **ROSE** package in R.

  Also, since the dataset is huge and consists of over 6 lakh records, the models would take a lot of time to run and their speed and performance decreases. Hence, we have taken a subset of the dataset with around 10,000 records randomly so that the chosen data is not biased and the model is trained on all classes of the output with better accuracy. Also, Cross Validation is performed for all the selected models to avoid overfitting and multicollinearity. traincontrol() function is used to perform cross-validation where the method is set to cv and number parameter is set to 5 which represents the number of folds for which cross-validation should take place. (Practical Guide to deal with Imbalanced Classification Problems in R, 2016)

```
summary(X_train$CASE_STATUS_NEW)
  0    1
993 7702
```

```
> summary(X_train_sample$CASE_STATUS_NEW)
   1    0
7702 7702
```

Data Modelling

After model resampling and cross-validation, we split the data set into train and test data for modelling the data and also understanding the accuracies and prediction of the model. Since the model has to predict a classification output, whether an applicant will be granted H-1B or not, we have chosen 4 models and trained them for the 80% of the dataset and verified the performances using the 20% of the dataset. Comparison of the accuracies, ROC curves and the AUC values generated by each model, their confusion matrix values are given below.

Understanding the Confusion Matrix

To attain good performance of a model and to improve its prediction power, we need to understand whether we need to reduce the False Positives or False Negatives for a model. Here, for example, an Applicant is denoted by X. From the definition of True Positives: Prediction is applicant X got the visa and Actually also visa was granted to X. Similarly, False Negatives imply that Prediction is applicant X did not receive the visa and Actually Visa was not granted to X. Hence, we understand that TruePositives and True Negatives are preferred scenarios. False Positives indicate that Prediction is applicant X will get the visa but actually, he did not get the Visa. False Negatives indicate that Prediction is applicant X will not get the visa but Actually, X was granted a visa. Hence, False Positives are more dangerous to our model compared to False Negatives. Our aim is to decrease the False Positives, with the cost of increasing False Negatives. Hence, we look at the specificity and recall values to maintain the balance between the FN and FPs.

Logistic Regression:

As the data that has been incorporated deals with a classification problem i.e. whether the application will be approved or denied. Logistic regression is practically good at solving these kinds of problems. Hence, a logistic regression model has been performed. A classic logistic regression model has been built with 306 predictors as One-Hot encoding was performed and the response variable as case status. In this model, 85% of accuracy was obtained on the original data and 61% of accuracy was obtained for the resampled data. The accuracy has been decreased when the prediction was run on the resampled data. Confusion matrix has shown the sensitivity of 0.181 and specificity of 0.944 with the imbalanced data.

Logistic Regression: Confusion matrix

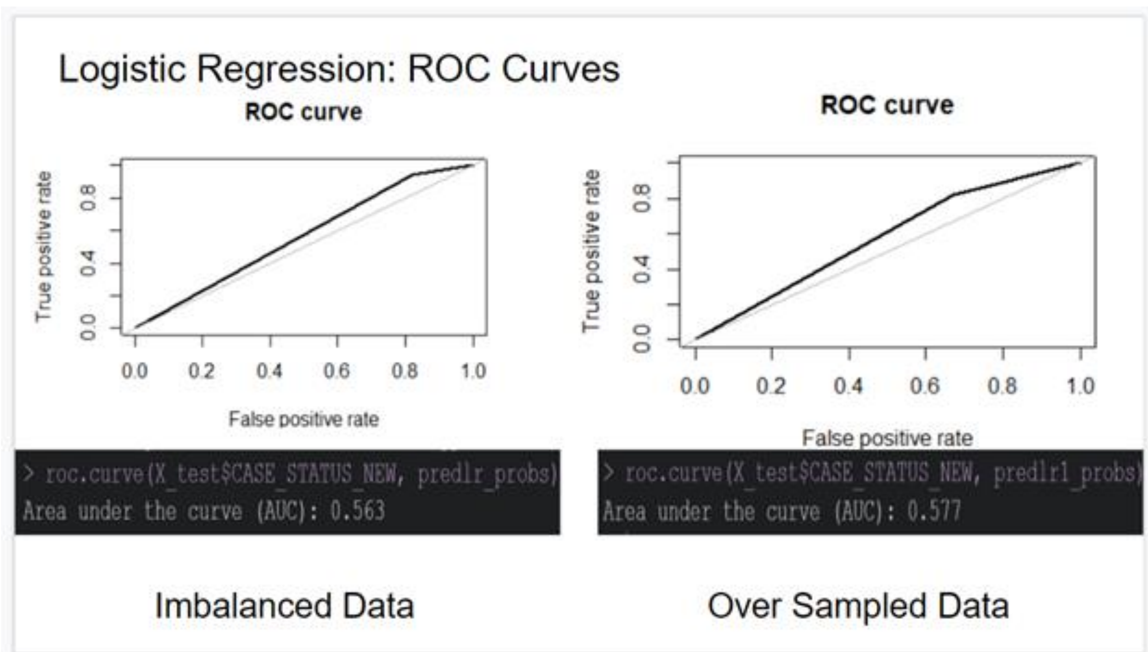
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0   44  108
         1  199 1823

               Accuracy : 0.8588
                 95% CI : (0.8434, 0.8732)
    No Information Rate : 0.8882
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1496

 Mcnemar's Test P-Value : 2.798e-07

            Sensitivity : 0.18107
            Specificity : 0.94407
```

Imbalanced Data

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0   43  638
         1  200 1293

               Accuracy : 0.6145
                 95% CI : (0.5937, 0.6351)
    No Information Rate : 0.8882
    P-Value [Acc > NIR] : 1

                  Kappa : -0.0858

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.17695
            Specificity : 0.66960
```

Over Sampled Data

Whereas with the resampled data sensitivity of 0.179 and specificity of 0.669 was obtained. It can be clearly observed that the false positive and true negative values are increased when trained on resampled data. Null deviance is 6028.2 and the residual deviance is 85567.6 these values are obtained with the actual data. The difference between null deviance and residual deviance is 79539.4.



The above figure depicts Roc curves for both actual data and the resampled data. The AUC values for actual data is 0.563 and the AUC value for resampled data is 0.577.

```
JOB_TITLE.SOFTWARE.DEVELOPMENT.ENGINEER                  -2.241e+14  2.477e+07   -9046274   <2e-16 ***
JOB_TITLE.SOFTWARE.DEVELOPMENT.ENGINEER.II                3.034e+15  2.091e+07  145070052   <2e-16 ***
JOB_TITLE.SOFTWARE.ENGINEER                              -7.713e+14  1.399e+07  -55134480   <2e-16 ***
JOB_TITLE.SOFTWARE.QUALITY.ASSURANCE.ENGINEER            -2.384e+14  2.170e+07  -10986522   <2e-16 ***
JOB_TITLE.SR..SOFTWARE.ENGINEER                           1.301e+15  1.880e+07   69225053   <2e-16 ***
JOB_TITLE.SYSTEMS.ANALYST                                -9.212e+14  1.588e+07  -57999225   <2e-16 ***
JOB_TITLE.SYSTEMS.ENGINEER                                1.736e+15  1.787e+07   97134408   <2e-16 ***
JOB_TITLE.TECHNICAL.LEAD                                 -1.472e+15  1.761e+07  -83556028   <2e-16 ***
JOB_TITLE.TECHNOLOGY.LEAD...US                            1.709e+13  1.696e+07    1007644   <2e-16 ***
JOB_TITLE.TECHNOLOGY.LEAD...US...PRACTITIONER                    NA         NA         NA       NA
WORKSITE_STATE.AK                                                NA         NA         NA       NA
WORKSITE_STATE.AL                                        -2.707e+15  7.129e+07  -37972610   <2e-16 ***
WORKSITE_STATE.AR                                        -2.940e+15  6.858e+07  -42861011   <2e-16 ***
WORKSITE_STATE.AZ                                        -2.855e+15  6.810e+07  -41923493   <2e-16 ***
WORKSITE_STATE.CA                                        -2.997e+15  6.783e+07  -44185918   <2e-16 ***
WORKSITE_STATE.CO                                        -3.020e+15  6.823e+07  -44266231   <2e-16 ***
WORKSITE_STATE.CT                                        -3.210e+15  6.815e+07  -47109675   <2e-16 ***
WORKSITE_STATE.DC                                        -1.794e+15  6.874e+07  -26103140   <2e-16 ***
WORKSITE_STATE.DE                                        -2.835e+15  6.846e+07  -41415985   <2e-16 ***
 [ reached getOption("max.print") -- omitted 55 rows ]
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance:  6028.2  on 8694  degrees of freedom
Residual deviance: 85567.6  on 8457  degrees of freedom
AIC: 86044

Number of Fisher Scoring iterations: 25

> |
```

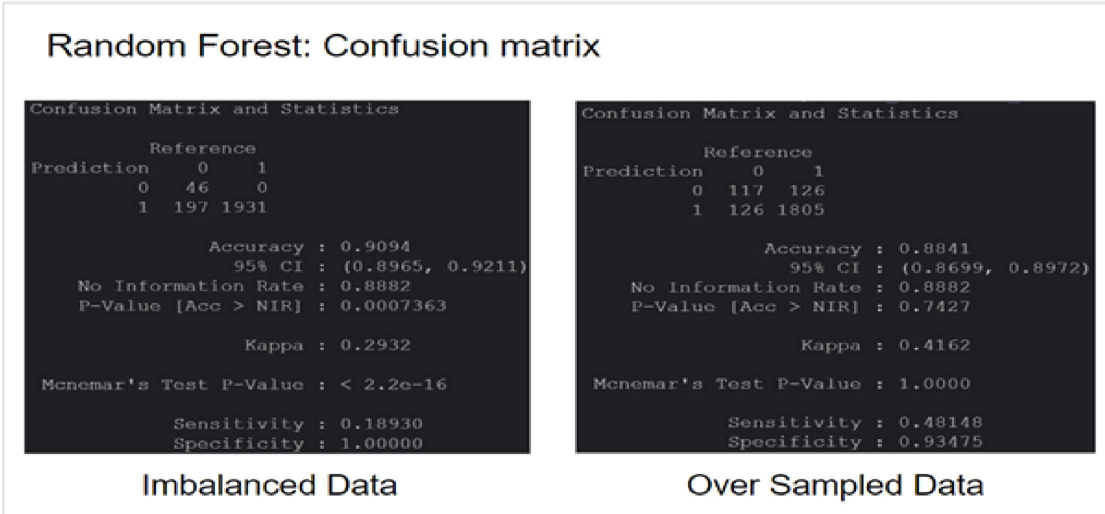
Null deviance is 21454 and the residual deviance is 369375 these values are obtained with the actual data. The difference between null deviance and residual deviance is 347921. Which shows that the model is performing well with the resampled data.

```
JOB_TITLE.MECHANICAL.ENGINEER                            -1.393e+15  1.877e+07  -74192028   <2e-16 ***
JOB_TITLE.NETWORK.ENGINEER                               -7.079e+14  1.726e+07  -41002789   <2e-16 ***
JOB_TITLE.OTHERS                                          2.624e+14  1.349e+07   19453841   <2e-16 ***
JOB_TITLE.PHYSICAL.THERAPIST                              2.423e+14  1.584e+07   15291816   <2e-16 ***
JOB_TITLE.PROGRAMMER.ANALYST                              1.281e+15  1.390e+07   92147250   <2e-16 ***
JOB_TITLE.PROJECT.ENGINEER                               -3.811e+15  2.025e+07 -188207681   <2e-16 ***
JOB_TITLE.PROJECT.MANAGER                                -3.711e+14  1.528e+07  -24294873   <2e-16 ***
JOB_TITLE.QA.ANALYST                                     -9.681e+13  1.825e+07   -5303842   <2e-16 ***
JOB_TITLE.RESEARCH.ASSOCIATE                              3.473e+14  1.564e+07   22201698   <2e-16 ***
JOB_TITLE.SENIOR.CONSULTANT                              -9.384e+14  1.610e+07  -58269367   <2e-16 ***
JOB_TITLE.SENIOR.SOFTWARE.DEVELOPER                      -6.328e+14  1.646e+07  -38438217   <2e-16 ***
JOB_TITLE.SENIOR.SOFTWARE.ENGINEER                        7.255e+14  1.432e+07   50657843   <2e-16 ***
JOB_TITLE.SENIOR.SYSTEMS.ANALYST.JC60                     2.335e+15  1.767e+07  132151736   <2e-16 ***
JOB_TITLE.SOFTWARE.DEVELOPER                             -1.360e+15  1.377e+07  -98758699   <2e-16 ***
JOB_TITLE.SOFTWARE.DEVELOPMENT.ENGINEER                   1.023e+15  1.995e+07   51264920   <2e-16 ***
JOB_TITLE.SOFTWARE.DEVELOPMENT.ENGINEER.II               -3.034e+15  2.067e+07 -146765964   <2e-16 ***
JOB_TITLE.SOFTWARE.ENGINEER                               1.115e+14  1.375e+07    8105293   <2e-16 ***
JOB_TITLE.SOFTWARE.QUALITY.ASSURANCE.ENGINEER             1.648e+15  1.748e+07   94259735   <2e-16 ***
JOB_TITLE.SR..SOFTWARE.ENGINEER                          -1.245e+15  1.780e+07  -69931320   <2e-16 ***
JOB_TITLE.SYSTEMS.ANALYST                                 6.770e+14  1.463e+07   46267083   <2e-16 ***
JOB_TITLE.SYSTEMS.ENGINEER                               -1.791e+15  1.715e+07 -104468612   <2e-16 ***
JOB_TITLE.TECHNICAL.LEAD                                  8.120e+14  1.537e+07   52815105   <2e-16 ***
JOB_TITLE.TECHNOLOGY.LEAD...US                           -2.508e+13  1.692e+07   -1482412   <2e-16 ***
JOB_TITLE.TECHNOLOGY.LEAD...US...PRACTITIONER                    NA         NA         NA       NA
WORKSITE_STATE.AK                                                NA         NA         NA       NA
WORKSITE_STATE.AL                                         4.253e+15  6.869e+07   61924168   <2e-16 ***
WORKSITE_STATE.AR                                         3.734e+15  6.819e+07   54754381   <2e-16 ***
WORKSITE_STATE.AZ                                         3.897e+15  6.788e+07   57411764   <2e-16 ***
WORKSITE_STATE.CA                                         3.916e+15  6.769e+07   57856709   <2e-16 ***
WORKSITE_STATE.CO                                         3.983e+15  6.793e+07   58635466   <2e-16 ***
WORKSITE_STATE.CT                                         3.949e+15  6.790e+07   58164497   <2e-16 ***
WORKSITE_STATE.DC                                         3.263e+15  6.840e+07   47711692   <2e-16 ***
WORKSITE_STATE.DE                                         4.102e+15  6.806e+07   60265706   <2e-16 ***
 [ reached getOption("max.print") -- omitted 55 rows ]
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance:  21454  on 15475  degrees of freedom
Residual deviance: 369375  on 15237  degrees of freedom
AIC: 369853

Number of Fisher Scoring iterations: 25

> |
```

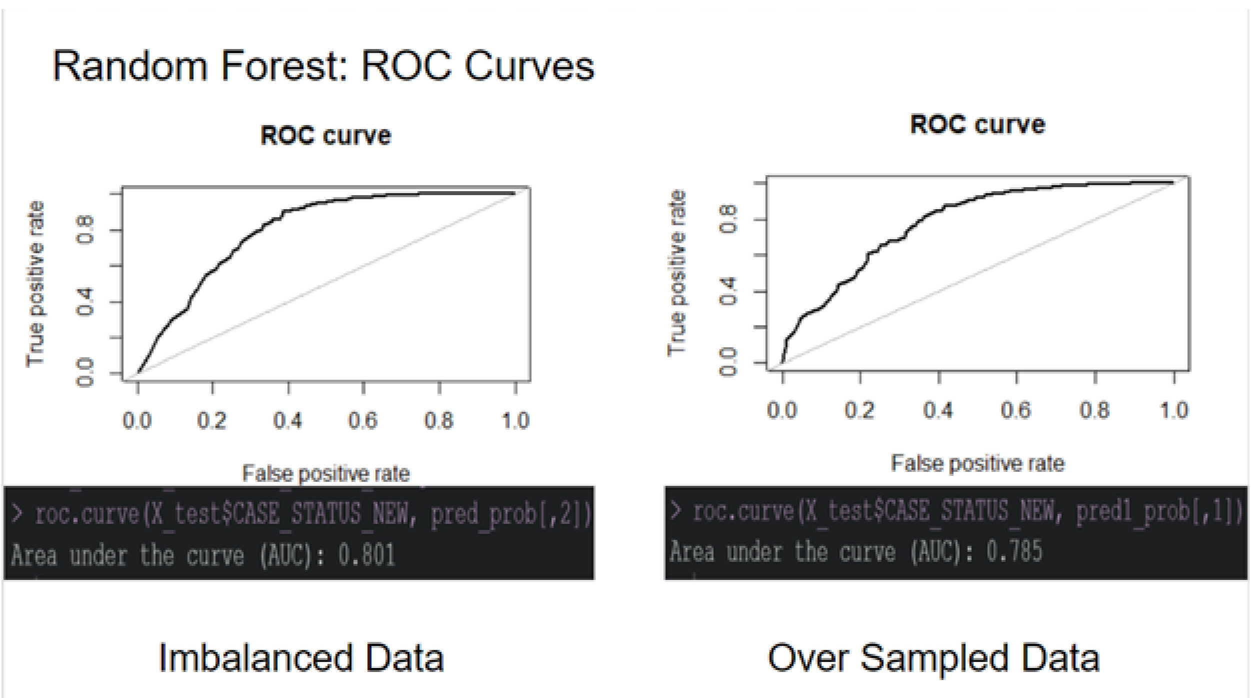
In this model both the accuracy and specificity is decreased but there is no significant change in the sensitivity when it is run on resampled data. From ROC Curves it is observed that AUC value doesn't have much difference. Hence in case of Logistic Regression, the model has performed better on actual data than on resampled data.

Random Forest:

In a quest to get a better prediction accuracy random forest has been implemented which is a classification model. It was performed on both the data sets i.e. actual and resampled. Confusion matrix has shown the sensitivity of 0.189 and specificity of 1.000 with the imbalanced data.



Whereas with the resampled data sensitivity of 0.481 and specificity of 0.934 was obtained. With the original data, an accuracy of 90% is obtained with zero false negatives & 197 false positives. The accuracy of 88% is achieved with the resampled data. False negatives are increased but the false positives are decreased to 126.



The above figure represents Roc Curves for the random forest model. It is observed that the model with actual data set has an AUC value of 0.801 and the model with resampled data has an AUC value of 0.785.

Support Vector Machine

By the concept of Support Vector Machine, it works by mapping the data to a high dimensional feature space so that data can be categorized. A separator between the categories is forced by the algorithm so that the data are transformed in such a way that the separator could be drawn as a hyperplane. Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we can maximize the classifier margin. Support vectors are the data points which help in building our model. The **"e1071"** package provides the SVM algorithm **svm()** function where the model is built using the 80 training data. Based on the trial and error method, we have chosen the kernel to "radial" and the Cost parameter as 1 and the gamma value to be 2. The decision values parameter is set to True to control the binary classifiers. The confusion matrix threshold is set as 0.5.

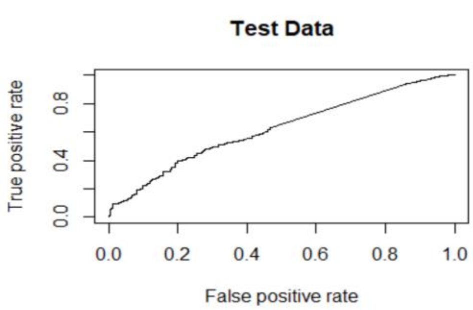
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0    5    0
         1  238 1931

               Accuracy : 0.8905
                 95% CI : (0.8766, 0.9033)
    No Information Rate : 0.8882
    P-Value [Acc > NIR] : 0.3827

                  Kappa : 0.036

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.02058
            Specificity : 1.00000
```

Imbalanced Data

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0    5    2
         1  238 1929

               Accuracy : 0.8896
                 95% CI : (0.8757, 0.9025)
    No Information Rate : 0.8882
    P-Value [Acc > NIR] : 0.4358

                  Kappa : 0.034

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.02058
            Specificity : 0.99896
```

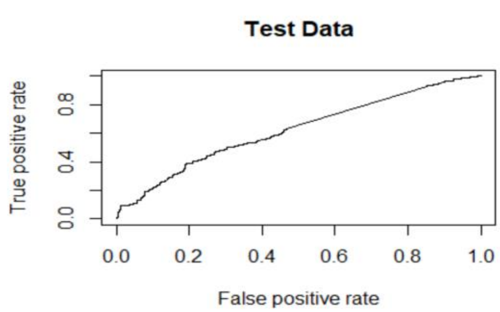
Over Sampled Data

From the above confusion matrix, in order to understand which data was providing us better accuracies by reducing the False Positives, for the imbalanced data we can see that the Specificity is 1 which is an ideal scenario and also False-negative value is zero. For the Over Sampled Data, we could see that specificity is close to 1. So, for both the data we could see that the False positives are reduced with an accuracy of 89% provided by the imbalanced data. From the below ROC curves and the respective Area under Curves (AUC) between imbalanced and oversampled data are seen. We could see that the AUC value was 62.1% with not much difference between the two data's and the ROC curve depicts the prediction power and better accuracy.




```
> auc$auc
Area under the curve: 0.6216
```

Imbalanced Data




```
> auc_sampled$auc
Area under the curve: 0.6214
```

Over Sampled Data

XG Boost

XGBoost is a scalable and accurate gradient boosting machine. This model is built solely to develop computational speed and model performance. The xgboost package provides us with the xgboost() model for which a new train and test data sets are required. Here we would store the label that the model has to predict in a separate variable. And the training input is taken in terms of the matrix. Here the objective parameter is set to binary: logistic since it has to train a binary classifier. The nrounds parameter is set to 250 so that the data is passed for 250 times. The class depth value is given as 5 indicating the tree maximum depth. Nthread parameter indicates the number of rounds the processor turns, here our model takes the nthread value as 6. The threshold for confusion matrix for prediction values is set to 0.5 and results in the below.

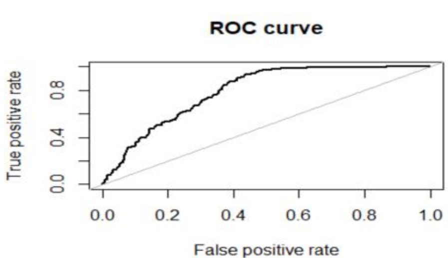
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0  123   51
         1  120 1880

               Accuracy : 0.9213
                 95% CI : (0.9092, 0.9323)
    No Information Rate : 0.8882
    P-Value [Acc > NIR] : 1.715e-07

                  Kappa : 0.5477

 Mcnemar's Test P-Value : 1.992e-07

            Sensitivity : 0.50617
            Specificity : 0.97359
```

Imbalanced Data

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0    0   89
         1  243 1842

               Accuracy : 0.8473
                 95% CI : (0.8315, 0.8622)
    No Information Rate : 0.8882
    P-Value [Acc > NIR] : 1

                  Kappa : -0.0637

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.00000
            Specificity : 0.95391
```

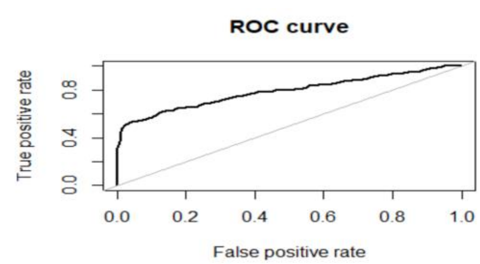
Over Sampled Data

From the above confusion matrices between the imbalanced data and oversampled data. Here we can observe that the imbalanced data has higher specificity where the False Positives are balanced when compared to oversampled data where the False positive values are increased. Hence the value of specificity is reduced which reduces the model performance. Hence, the model accuracy is 92% for the imbalanced data over the oversampled data accuracy. From the below ROC curves, we can see that better ROC curves are generated for imbalanced data with better AUC value when compared to the oversampled data. Here the AUC value is 0.796 which is comparatively a little better when calculated for the oversampled data.




```
> roc.curve(Y_test, xgb_pred_probs)
Area under the curve (AUC): 0.796
```

Imbalanced Data




```
> roc.curve(Y_test, xgb_pred_probs1)
Area under the curve (AUC): 0.790
```

Over Sampled Data

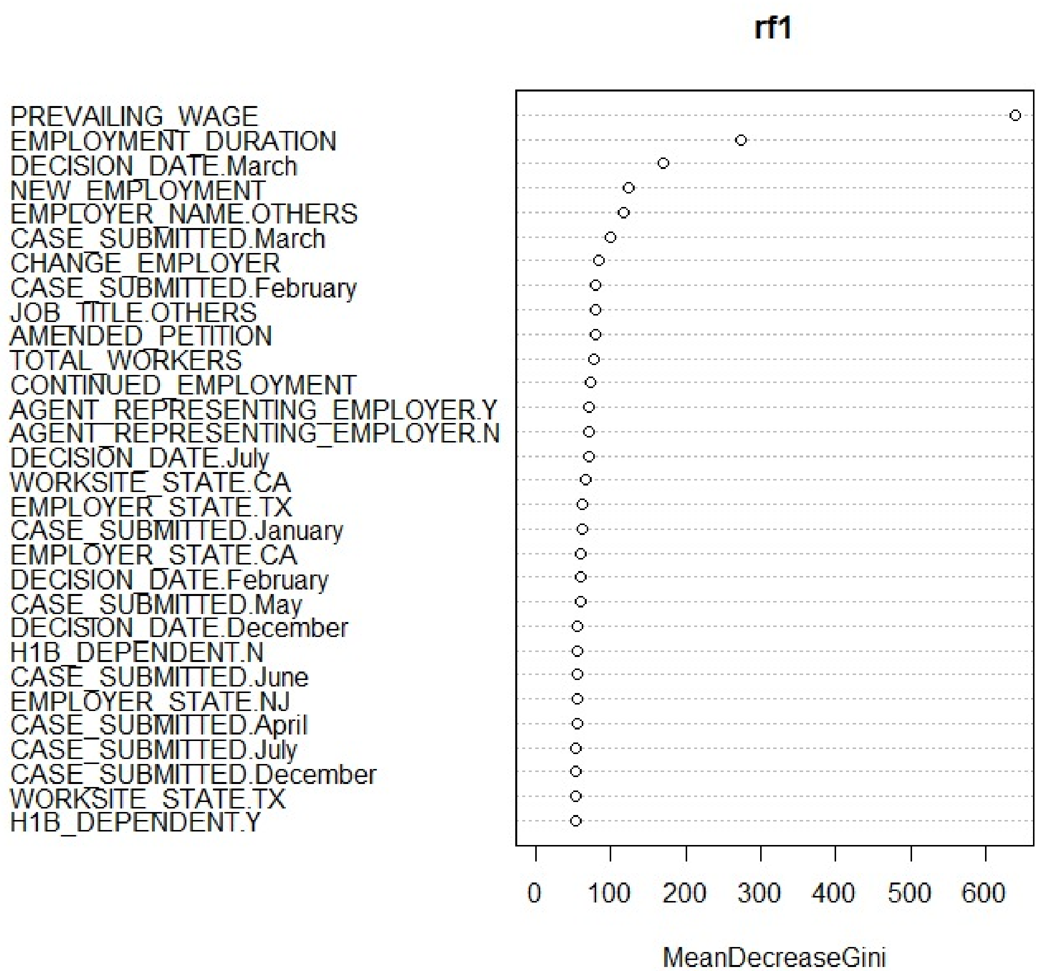
(XGBoost R Tutorial, n.d.)

Model Comparison

From the below accuracies, we can say that all the model performances are efficient in terms of accuracies. Hence, we also consider the AUC values while choosing the best model for the given dataset, as the AUC values compare the model's prediction power using various threshold values. Hence from the below list of values, we can interpret that tree-based models outperform logistic and SVM.

Model	Accuracy	AUC	Accuracy_sampled	AUC_sampled
Logistic Regression	85.88%	56.3%	61.45%	57.7%
SVM	89.05%	62.16%	88.96%	62.14%
Random Forest	90.94%	80.01%	88.41%	78.5%
XGBoost	92.13%	79.6%	84.73%	79%

From the below variable importance plot, we can understand the significant variables to predict the visa case status for an applicant. We could see that prevailing wage, duration of employment, job roles and the status decision outcomes which are released in March month, worksite location, etc are the most significant factors responsible for predicting the approval of the visa.



Conclusion

Working towards this project, we learned the importance of Data cleaning and preprocessing. Understanding that data preprocessing step plays a major role in the data and also on the model output predicted. Implementing methods like one-hot encoding, cross-validation and resampling of the data for better performance of the model and understanding the changes it had made to the output. By analyzing the columns, and deciding between which columns are significant and which columns are uninformative makes an impact on the model. This is a very important stage while identifying significant predictors. Choosing the sample of data so that the model can be trained on unbiased data to provide an unbiased model.

While working on the models, we learned the concepts of tuning the model and its importance. Based on the kind of output the model has to generate and the independent variables we had, to identify the significant model was an important stage. By trial and error method, the various parameters of the model are tuned and model training is done. We also implemented the ROC and AUC values and analyzed the importance of these concepts. Since the project deals with a real-life problem, we could predict that factors like an applicant's wage, the employer company, whether the applicant is a full-time employee or not, the work location etc are few of the significant factors for predicting whether an applicant will be granted with the H-1B visa or not.

References

Anbarasan, A. (2019, May 15). *H1B Prediction*. From kaggle: https://www.kaggle.com/abishekanbarasan1995/h1b-case-status-prediction#H-1B_Disclosure_RAW_Data_FY18.csv

Jhanji, D. (2018, June 20). *Predicting the Status of H-1B Visa Applications*. From datacamp: https://www.datacamp.com/community/tutorials/predicting-H-1B-visa-status-python

Practical Guide to deal with Imbalanced Classification Problems in R. (2016, March 28). From Analytics Vidhya: https://www.analyticsvidhya.com/blog/2016/03/practical-guide-deal-imbalanced-classification-problems/

XGBoost R Tutorial. (n.d.). From XGBoost: https://xgboost.readthedocs.io/en/latest/R-package/xgboostPresentation.html

Appendix:

Data Field	Description
Case Number	Each unique number of identified visa case
Case Status	Represents whether the visa is certified or denied
Case Submitted Date	Date of visa case submission
Decision Date	Date of the visa case decision
Visa Class	Represents visa class
Employer Name	Represents the name of the Employer
Employer Country	Represents employer’s country
Employer Duration	Represents the duration of the employment
New Employment	Represents whether the applicant is newly employed
Change Employer	Represents whether the applicant has changed his job
Worksite	Represents the location of work
SOC Code	Represents code of the Job Role
SOC Name	Represents the name of the Job Role
Job Title	Represents the name of the Job Role
Agent Representing Employer	Represents whether the applicant represents an agent
Prevailing Wage	Represents salary with benefits
Wage Rate	Represents salary

CODE

Packages Installed

```
library(tidyverse)
library(fastDummies)
library(lubridate)
library(dmm)
library(ROCR)
library(MLmetrics)
library(dplyr)
library(tidyr)
library(caret)
library(funModeling)
library(ROSE)
library(plotly)
library(dplyr)
library(tidyr)
library(corrplot)
library(ggplot2)
require(scales)
library(doParallel)
library(randomForest)
library(e1071)
library(xgboost)
```

Data loading and filtration

```
df <- read.csv("C:\\Users\\samsu\\Desktop\\Spring 2020\\OR 568\\Final Project\\H-1B_Disclosure_RAW_Data_FY18.csv")
USADf = subset(df, EMPLOYER_COUNTRY == "UNITED STATES OF AMERICA" & VISA_CLASS == 'H-1B', select=ï..CASE_NUMBER:ORIGINAL_CERT_DATE)
USADf = subset(USADf, select = -c(ï..CASE_NUMBER,EMPLOYER_BUSINESS_DBA,EMPLOYER_COUNTRY,EMPLOYER_ADDRESS,EMPLOYER_CITY,EMPLOYER_COUNTRY,

EMPLOYER_POSTAL_CODE,EMPLOYER_PROVINCE,EMPLOYER_PHONE,EMPLOYER_PHONE_EXT,AGENT_ATTORNEY_NAME,

AGENT_ATTORNEY_CITY,AGENT_ATTORNEY_STATE,SUPPORT_H1B,LABOR_CON_AGREE,PUBLIC_DISCLOSURE_LOCATION,

ORIGINAL_CERT_DATE,PW_SOURCE_YEAR,PW_SOURCE_OTHER,PW_SOURCE,WORKSITE_CITY,WORKSITE_COUNTY,
```

```
PW_WAGE_LEVEL,WORKSITE_POSTAL_CODE,WAGE_RATE_OF_PAY_FROM,WAGE_RATE_OF_PAY_TO,VISA_CLASS,
                    WAGE_UNIT_OF_PAY,SOC_CODE,NAICS_CODE))
```

Data Visualization

```
data<- read.csv("C:/Users/gunaganti meghana/Desktop/GMU/COURSES/Spring 2020 OR 568/Assignments/FINAL PROJECT/H-1B_Disclosure_Data_18.csv")

data = subset(data, select = -c(NAICS_CODE))
PREVAILING_WAGE <- as.numeric(gsub(",", "",as.character(data$PREVAILING_WAGE)))
data$PREVAILING_WAGE = ifelse(data$PW_UNIT_OF_PAY=='Bi-Weekly',data$PREVAILING_WAGE <- (PREVAILING_WAGE/2)*52.143,
                    ifelse(data$PW_UNIT_OF_PAY=='Hour',data$PREVAILING_WAGE <- (PREVAILING_WAGE*40)*52.143,
                        ifelse(data$PW_UNIT_OF_PAY=='Month',data$PREVAILING_WAGE <- PREVAILING_WAGE*12,

ifelse(data$PW_UNIT_OF_PAY=='Week',data$PREVAILING_WAGE <- PREVAILING_WAGE*52.143,

ifelse(data$PW_UNIT_OF_PAY=='Year',data$PREVAILING_WAGE <- PREVAILING_WAGE,NA
                                )))))
data<- data%>%
  filter(VISA_CLASS == 'H-1B')
as.data.frame(data)
head(data)
colnames(data)

#Retrieving Numeric Data

NumericData<-select_if(data, is.numeric)
as.data.frame(NumericData)
drop_na(NumericData)
NumericData[!(is.na(NumericData=="")),]

#Correlation Plot

ND_CorrelationPlot <- cor(NumericData)
```

```
corrplot(ND_CorrelationPlot, method = "circle")

#Group by Case Status and sum of each case status
CASE_STATUS_GROUP <- group_by(data,CASE_STATUS)
CASE_STATUS_GROUP
CASE_STATUS_GROUP_Summary<-
data.frame(table(CASE_STATUS_GROUP$CASE_STATUS))
as.data.frame(CASE_STATUS_GROUP_Summary)
#Plot Count of CaseStatus
COUNT_CASE_STATUS_PLOT<-ggplot(CASE_STATUS_GROUP_Summary, aes(Var1,
Freq))+ geom_bar(stat='identity') + scale_y_continuous(name="Count", labels = comma)
COUNT_CASE_STATUS_PLOT

COUNT_CASE_STATUS_PLOT_O<-ggplot(CASE_STATUS_GROUP_Summary,
aes(x=reorder(Var1, -Freq),y=Freq))+ geom_bar(stat='identity',fill="#FF9999", colour="black")
+ scale_y_continuous(name="Count", labels = comma)+xlab("Case Status") +
  ggtitle("Count of Case Status")
COUNT_CASE_STATUS_PLOT_O

#Groupby Jobtitle, Prevailing Wage with an average of it.
JT_PW_Group <- group_by(data,PREVAILING_WAGE,JOB_TITLE)
JT_PW_Group
as.data.frame(JT_PW_Group)
JT_PW_Group_Summary<-summarize(JT_PW_Group,Average_Prevailing_Wage=
mean(as.numeric(PREVAILING_WAGE),na.rm = TRUE))
is.null(JT_PW_Group_Summary)
as.data.frame(JT_PW_Group_Summary)
JT_PW_Group_Summary_Desc<-
JT_PW_Group_Summary[order(JT_PW_Group_Summary$Average_Prevailing_Wage,
rev(JT_PW_Group_Summary$JOB_TITLE), decreasing = TRUE), ]
JT_PW_Group_Summary_Top20<-head(JT_PW_Group_Summary_Desc,20)
as.data.frame(JT_PW_Group_Summary_Top20)

#Plot Top 20 Job Title with their Average Prevailing Wage

Top20_JT_PW_plot<-ggplot(JT_PW_Group_Summary_Top20, aes(x=JOB_TITLE,y =
PREVAILING_WAGE)) +geom_bar(stat = "identity") + scale_y_continuous(name="Top 20
Average Prevailing Wages of Job Title", labels =
comma)+theme(axis.text.x=element_text(angle=90,hjust=1,vjust=0.5))+coord_flip()
Top20_JT_PW_plot
```

```
Top20_JT_PW_plot_O<-ggplot(JT_PW_Group_Summary_Top20, aes(x=reorder(JOB_TITLE,-
PREVAILING_WAGE),y=PREVAILING_WAGE)) +geom_bar(stat =
"identity",fill="#E69F00", colour="black") + scale_y_continuous(name="Prevailing Wage",
labels = comma)+theme(axis.text.x=element_text(angle=90,hjust=1,vjust=0.5))+coord_flip() +
  xlab("Job Title") +
  ggtitle("Top 20 Average Prevailing Wages of Job Title")
Top20_JT_PW_plot_O
#Groupby Case Status, Employer Name and sum by its each case status group .
EN_PW_Group <- group_by(data,EMPLOYER_NAME)
EN_PW_Group

EN_PW_Group_Summary <- data.frame(table(EN_PW_Group$EMPLOYER_NAME))
EN_PW_Group_Summary_Desc <-
EN_PW_Group_Summary[order(EN_PW_Group_Summary $Freq,
rev(EN_PW_Group_Summary $Var1),decreasing = TRUE),]
EN_PW_Group_Summary_Top20 <- head(EN_PW_Group_Summary_Desc,20)
as.data.frame(EN_PW_Group_Summary_Top20)

#Plot Top 20 Employers
Top20_EN<-ggplot(EN_PW_Group_Summary_Top20, aes(Var1,Freq))+
geom_bar(stat='identity') + scale_y_continuous(name="Top 20 Employer Names", labels =
comma)+theme(axis.text.x=element_text(angle=90,hjust=1,vjust=0.5))+coord_flip()
Top20_EN

Top20_EN_O<-ggplot(EN_PW_Group_Summary_Top20, aes(x=reorder(Var1,-Freq),y=Freq))+
geom_bar(stat='identity',fill="#D55E00", colour="black") + scale_y_continuous(name="Count",
labels = comma)+theme(axis.text.x=element_text(angle=90,hjust=1,vjust=0.5))+coord_flip()+
xlab("Employer Name") +
  ggtitle("Top 20 Employer Names")
Top20_EN_O


#Plot Top 20 SOC Names with their Average Prevailing Wage
SN_PW_Group <- group_by(data,SOC_NAME,PREVAILING_WAGE)
SN_PW_Group
SN_PW_Group_Summary<-summarize(SN_PW_Group,Average_PrevailingWage =
mean(as.numeric(PREVAILING_WAGE),na.rm = TRUE))
as.data.frame(SN_PW_Group_Summary)
```

```
SN_PW_Group_Summary_Desc<-
SN_PW_Group_Summary[order(SN_PW_Group_Summary$Average_PrevailingWage,
rev(SN_PW_Group_Summary$SOC_NAME), decreasing = TRUE), ]
SN_PW_Group_Summary_Top20<-head(SN_PW_Group_Summary_Desc,20)
as.data.frame(SN_PW_Group_Summary_Top20)

#Plot Top 20 SOC Names with their Average Prevailing Wage
Top20_SN<-ggplot(SN_PW_Group_Summary_Top20,
aes(SOC_NAME,Average_PrevailingWage ))+ geom_bar(stat='identity') +
scale_y_continuous(name="Top 20 Average Prevailing Wages of Soc Name", labels =
comma)+theme(axis.text.x=element_text(angle=90,hjust=1,vjust=0.5))+coord_flip()
Top20_SN

Top20_SN_O<-ggplot(SN_PW_Group_Summary_Top20)+
geom_bar(aes(x=reorder(SOC_NAME,Average_PrevailingWage),y=Average_PrevailingWage
),stat='identity',fill="#0072B2", colour="black") +theme_minimal()+
scale_y_continuous(name="Prevailing Wage", labels =
comma)+theme(axis.text.x=element_text(angle=90,hjust=1,vjust=0.5))+coord_flip()+xlab(" Soc
Names") +
  ggtitle("Top 20 Average Prevailing Wages of Soc Name")
Top20_SN_O

#Filtering only CERTIFIED and DENIED Case Status
data_boxplot <- data%>%
  filter(CASE_STATUS == 'CERTIFIED' | CASE_STATUS == 'DENIED')

# plot boxplot
ggplot(aes(y = PREVAILING_WAGE, x = CASE_STATUS, fill = CASE_STATUS,
           notch = TRUE, notchwidth = .3),
       data = data_boxplot) +
  geom_boxplot(notch = TRUE) +
  scale_fill_manual(values = c("#29a329", "#ea4b1f"))+
  scale_y_continuous(limits = c(0, 150000),
                     breaks = seq(0, 150000, 5000)) +
  ggtitle("Wages for certified & denied H1B cases")+
  theme(
    plot.title = element_text(size = rel(2)),
    panel.background = element_rect(fill = 'light gray'),
    panel.grid.major = element_line(colour = '#f0f0f0'),
    panel.grid.major.x = element_line(linetype = 'blank'),
```

```
    panel.grid.minor = element_line(linetype = 'blank')
  )
```

Data preprocessing

Near-Zero Variance

```
#Detecting near zero variance
x = nearZeroVar(USADf, saveMetrics = TRUE)
str(x, vec.len=2)
x[x[,"zeroVar"] > 0, ]
x[x[,"zeroVar"] + x[,"nzv"] > 0, ]
```

Handling missing values

```
#Include original case_status if needed
#To find the NA's and missing values
summary(USADf)
str(USADf)
USADf[USADf==""]<-NA
sapply(USADf, function(x) sum(is.na(x)))

USADf = na.omit(USADf)
sapply(USADf, function(x) sum(is.na(x)))

head(USADf)
dim(USADf)
```

Calculating and Formatting the columns

```
#Adding new column Duration of work and dropping employment start and end dates.
USADf$EMPLOYMENT_END_DATE = as.Date(USADf$EMPLOYMENT_END_DATE,
"%m/%d/%Y")
USADf$EMPLOYMENT_START_DATE = as.Date(USADf$EMPLOYMENT_START_DATE,
"%m/%d/%Y")
USADf$EMPLOYMENT_DURATION <- as.numeric(USADf$EMPLOYMENT_END_DATE-
USADf$EMPLOYMENT_START_DATE)
USADf$EMPLOYMENT_DURATION <- round(USADf$EMPLOYMENT_DURATION*0.033)

USADf = subset(USADf, select = -c(EMPLOYMENT_END_DATE,EMPLOYMENT_START_DATE))
```

Scaling

```
#Considering only months
USADf$DECISION_DATE = as.Date(USADf$DECISION_DATE, "%m/%d/%Y")
USADf$CASE_SUBMITTED = as.Date(USADf$CASE_SUBMITTED, "%m/%d/%Y")

USADf$CASE_SUBMITTED <- format(USADf$CASE_SUBMITTED,'%B')
USADf$CASE_SUBMITTED <- as.factor(USADf$CASE_SUBMITTED)

USADf$DECISION_DATE <- format(USADf$DECISION_DATE,'%B')
USADf$DECISION_DATE <- as.factor(USADf$DECISION_DATE)

#adding binary Response variable
USADf$CASE_STATUS_NEW = ifelse(USADf$CASE_STATUS=='CERTIFIED',1,0)
USADf$CASE_STATUS_NEW <- as.factor(USADf$CASE_STATUS_NEW)

USADf = subset(USADf, select = -c(CASE_STATUS))

#Converting the pay to anually
PREVAILING_WAGE <- as.numeric(gsub(",", "",as.character(USADf$PREVAILING_WAGE)))
USADf$PREVAILING_WAGE = ifelse(USADf$PW_UNIT_OF_PAY=='Bi-
Weekly',USADf$PREVAILING_WAGE <- (PREVAILING_WAGE/2)*52.143,
                    ifelse(USADf$PW_UNIT_OF_PAY=='Hour',USADf$PREVAILING_WAGE <-
(PREVAILING_WAGE*40)*52.143,
                        ifelse(USADf$PW_UNIT_OF_PAY=='Month',USADf$PREVAILING_WAGE
<- PREVAILING_WAGE*12,

ifelse(USADf$PW_UNIT_OF_PAY=='Week',USADf$PREVAILING_WAGE <-
PREVAILING_WAGE*52.143,

ifelse(USADf$PW_UNIT_OF_PAY=='Year',USADf$PREVAILING_WAGE <-
PREVAILING_WAGE,NA
                                )))))
#Dropping PW unit Unit of pay variables
USADf = subset(USADf, select = -c(PW_UNIT_OF_PAY))

summary(USADf)
```

Refactoring

```
##Refactor the below columns to 50 levels
x = freq(data=USADf$JOB_TITLE)
job = x$var[1:50]

USADf$JOB_TITLE = ifelse(USADf$JOB_TITLE %in%
c(job),as.character(USADf$JOB_TITLE),"OTHERS")
USADf$JOB_TITLE = as.factor(USADf$JOB_TITLE)

y = freq(data=USADf$SOC_NAME)
socname = y$var[1:50]

USADf$SOC_NAME = ifelse(USADf$SOC_NAME %in%
c(socname),as.character(USADf$SOC_NAME),"OTHERS")
USADf$SOC_NAME = as.factor(USADf$SOC_NAME)

z = freq(data=USADf$EMPLOYER_NAME)
emp = z$var[1:50]

USADf$EMPLOYER_NAME = ifelse(USADf$EMPLOYER_NAME %in%
c(emp),as.character(USADf$EMPLOYER_NAME),"OTHERS")
USADf$EMPLOYER_NAME = as.factor(USADf$EMPLOYER_NAME)

summary(USADf)
str(USADf)
```

One-Hot Encoding

```
encode =
dummyVars(~CASE_SUBMITTED+DECISION_DATE+EMPLOYER_NAME+EMPLOYER_STATE+
JOB_TITLE+SOC_NAME+WORKSITE_STATE+

AGENT_REPRESENTING_EMPLOYER+FULL_TIME_POSITION+H1B_DEPENDENT+WILLFUL_
VIOLATOR, data = USADf)
USADf_oh = as.data.frame(predict(encode, newdata = USADf))

USADf = subset(USADf, select = -
c(CASE_SUBMITTED,DECISION_DATE,EMPLOYER_NAME,EMPLOYER_STATE,JOB_TITLE,S
OC_NAME,

WORKSITE_STATE,AGENT_REPRESENTING_EMPLOYER,FULL_TIME_POSITION,H1B_DEPE
NDENT,
                WILLFUL_VIOLATOR))
final <- cbind(USADf,USADf_oh)
```

```
final = subset(final, select = -
c(WORKSITE_STATE.,EMPLOYER_STATE.,AGENT_REPRESENTING_EMPLOYER.,FULL_TIM
E_POSITION.,
                          H1B_DEPENDENT.,WILLFUL_VIOLATOR.))

#Remove spaces from column names
names(final) = make.names(names(final))
dim(final)
summary(final)
head(final)
```

Cross Validation and Resampling

```
class(final)
summary(final$CASE_STATUS_NEW)

#random subset of around 10000 observations
set.seed(123)
index = sample(nrow(final),nrow(final)*.017)
final_test = final[index,]
summary(final_test$CASE_STATUS_NEW)
dim(final_test)

#Train Test Split for unsampled data
set.seed(123)
trainIndex = sample(nrow(final_test),nrow(final_test)*.8)
X_train = final_test[trainIndex,]
X_test = final_test[-trainIndex,]


#Cross validation
train = trainControl(method="cv", number=5)
lrcv = train(CASE_STATUS_NEW ~.,data=X_train,trControl=train,method="glm")
rfcv = train(CASE_STATUS_NEW ~.,data=X_train,trControl=train,method="rf")
svmcv = train(CASE_STATUS_NEW
~.,data=X_train,trControl=train,method="svmRadial",scale=FALSE)
lrcv
plot(rfcv)
plot(svmcv)

summary(X_train$CASE_STATUS_NEW)
summary(X_test$CASE_STATUS_NEW)
#Over sampling(run the models and capture the details after and before over sampling)
```

```
balanced_over = ovun.sample(CASE_STATUS_NEW ~ .,data = X_train, method = "over", N = 15476,
seed = 1)$data
table(balanced_over$CASE_STATUS_NEW)
dim(balanced_over)
X_train_sample = balanced_over
```

Modelling the Data, Confusion Matrix and ROC curves

Logistic Regression

```
#Modelling
#Logistic Regression
logit <- glm(CASE_STATUS_NEW ~., data = X_train, family = binomial)
summary(logit)
predlr_probs = predict(logit, X_test[-10],type ='response')
predlr = ifelse(predlr_probs>0.5,1,0)
confusionMatrix(as.factor(predlr),X_test$CASE_STATUS_NEW)
roc.curve(X_test$CASE_STATUS_NEW, predlr_probs)


logit1 <- glm(CASE_STATUS_NEW ~., data = X_train_sample, family = binomial)
summary(logit1)
predlr1_probs = predict(logit1, X_test[-10],type ='response')
predlr1 = ifelse(predlr1_probs>0.5,1,0)
confusionMatrix(as.factor(predlr1),X_test$CASE_STATUS_NEW)
roc.curve(X_test$CASE_STATUS_NEW, predlr1_probs)
```

Random Forest

```
#Modelling in RF (Our aim is to decrease the number of false positives as we dont want to give false
hope
#to a person saying they will get an H1B Visa but eventually they won't.)
library(doParallel)
detectCores()
c <- makePSOCKcluster(6)
registerDoParallel(c)
stopCluster(c)


library(randomForest)
rf <- randomForest(CASE_STATUS_NEW ~.,data=X_train)
summary(rf)
pred = predict(rf, X_test[-10])
```

```
pred_prob = predict(rf, X_test[-10], type="prob")
confusionMatrix(pred,X_test$CASE_STATUS_NEW)
#a$overall
varImpPlot(rf)
importance(rf)
plot(rf)
#F1_Score(X_test$CASE_STATUS_NEW,pred)
roc.curve(X_test$CASE_STATUS_NEW, pred_prob[,2])

rf1 <- randomForest(CASE_STATUS_NEW ~.,data=X_train_sample)
summary(rf1)
pred1 = predict(rf1, X_test[-10])
pred1_prob = predict(rf1, X_test[-10], type="prob")
confusionMatrix(pred1,X_test$CASE_STATUS_NEW)
varImpPlot(rf1)
importance(rf1)
plot(rf1)
roc.curve(X_test$CASE_STATUS_NEW, pred1_prob[,1])
```

SVM

```
#SVM
library(e1071)

rocplot=function(pred, truth, ...){
  predob = prediction(pred, truth)
  perf = performance(predob, "tpr", "fpr")
  plot(perf,...)}

svmfit=svm(CASE_STATUS_NEW ~.,data = X_train, kernel = "radial", scale = FALSE, cost= 1,
gamma=2,
          decision.values=T)
summary(svmfit)
ypred=predict(svmfit, X_test[-10])
fitted=attributes(predict(svmfit, X_test[-10],decision.values=TRUE))$decision.values
confusionMatrix(ypred,X_test$CASE_STATUS_NEW)
rocplot(fitted,X_test$CASE_STATUS_NEW,main="Training Data")

svmfit1=svm(CASE_STATUS_NEW ~.,data = X_train_sample, kernel = "radial", scale = FALSE,cost=
1, gamma=2,
           decision.values=T)
summary(svmfit1)
ypred1=predict(svmfit1, X_test[-10])
fitted1=attributes(predict(svmfit1, X_test[-10],decision.values=TRUE))$decision.values
```

```
confusionMatrix(ypred1,X_test$CASE_STATUS_NEW)
rocplot(fitted1,X_test$CASE_STATUS_NEW,main="Training Data")
```

XGBoost

```
#XGBoost
library(xgboost)
X_train_new = subset(X_train, select = -c(CASE_STATUS_NEW))
X_test_new = subset(X_test, select = -c(CASE_STATUS_NEW))
Y_train = as.numeric(X_train$CASE_STATUS_NEW) - 1
Y_test = X_test$CASE_STATUS_NEW

xgb = xgboost(data = as.matrix(X_train_new),label=as.matrix(Y_train),max.dclassepth=5,eta = 1,
        nthread=6,nrounds=250,objective = "binary:logistic")
xgb_pred_probs = predict(xgb,as.matrix(X_test_new))
xgb_pred = ifelse(xgb_pred_probs > 0.5,1,0)
confusionMatrix(as.factor(xgb_pred), as.factor(Y_test))
roc.curve(Y_test, xgb_pred_probs)


X_train_sample_new = subset(X_train_sample, select = -c(CASE_STATUS_NEW))
Y_train_sample = as.numeric(X_train_sample$CASE_STATUS_NEW) - 1
#X_train_sample_new = subset(X_train_sample, select = -c(CASE_STATUS_NEW))
#Y_train_sample = as.numeric(X_train_sample$CASE_STATUS_NEW) - 1

xgb_sample = xgboost(data = as.matrix(X_train_sample_new),label=as.matrix(Y_train_sample),
            max.dclassepth=5,eta = 1,nthread=6,nrounds=200,objective = "binary:logistic")
xgb_pred_probs1 = predict(xgb_sample,as.matrix(X_test_new))
xgb_pred1 = ifelse(xgb_pred_probs1 > 0.00001,1,0)
confusionMatrix(as.factor(xgb_pred1), as.factor(Y_test))
roc.curve(Y_test, xgb_pred_probs1)
summary(xgb_pred_probs1)
#Feature selection
set.seed(100)
rPartMod <- train(CASE_STATUS_NEW ~.,data=USADf, method="rpart")
varImp(rPartMod)
```