H1B CASE STATUS PREDICTION

Authors:

Jinal Jain jjjain@wpi.edu (MS Data Science, WPI)
Janvi Kothari jkkothari@wpi.edu (MS Data Science, WPI)
Mihin Sumaria mssumaria@wpi.edu (MS Data Science, WPI)
Mihir Sawant msawant@wpi.edu (MS Data Science, WPI)
Rushikesh Naidu ranaidu@wpi.edu (MS Data Science, WPI)

Abstract

The **H-1B** is a visa in the United States under the Immigration and Nationality Act, section which allows U.S. employers to employ foreign workers in specialty occupations. H-1B visa class is among the most sought after visa-categories. The applications for H-1B visa differ in many ways: the company name, prevailing wages, workplace location, the type of job, job title, year of petition and alike. In this report, we attempt to predict the status of the visa petition based on the visa-application metadata. The intent of this study is to discover how visa status outcome is influenced by attributes of user application. The classifier designed in this report could be utilized by both, H-1B 1 aspirants and employers, to gauge the likelihood of visa certification, before and after filing the petition.

1. Introduction

What is H-1B Visa?

The H-1B program allows companies in the United States to temporarily employ foreign workers in occupations that require the theoretical and practical application of a body of highly specialized knowledge and a bachelors degree or higher in the specific specialty, or its equivalent. H-1B specialty occupations may include fields such as science, engineering and information technology.

In carrying out its responsibility for the processing of labor certification and labor attestation applications, the Office of Foreign Labor Certification (OFLC) generates program data that is essential both for internal assessment of program effectiveness and for providing the Department's external stakeholders with useful information about the immigration programs administered by OFLC. In line with the Department's commitment to the Open Government initiative and specific regulatory disclosure requirements, OFLC makes public the annual releases of program disclosure data to assist with external research and program evaluation.

We are interested in learning whether based on a few variables like application date, wage, locations, etc can the decision for an applicant be predicted. We pulled out the H-1B filing data for FY 2017 from the Office of Foreign Labor Certification.

We also tried to infer from this whether H-1B visa allocation is truly a lottery or random process or not.

2. Method

2.1 Data Gathering

The data was downloaded from the website of Office of Foreign Labor Certification: https://www.foreignlaborcert.doleta.gov/pdf/PerformanceData/2017/H-1B_Disclosure_Data_FY17.xlsx. The data was available for previous years but we chose the FY 2017 data for our analysis.

2.2 Data Description

The file is available and was downloaded in xlsx format. The total size of the file is 245.285MB. We have 645,241 number of observations in total with 52 variables, out of which 29 are categorical variables and 23 are numerical variables. The file structure and description is as follows:

FIELD NAME	DESCRIPTION
CASE_NUMBER	Unique identifier assigned to each application submitted for processing to the Chicago National Processing Center.
CASE_STATUS	Status associated with the last significant event or decision. Valid values include “Certified,” “Certified-Withdrawn,” Denied,” and “Withdrawn”.
CASE_SUBMITTED	Date and time the application was submitted.
DECISION_DATE	Date on which the last significant event or decision was recorded by the Chicago National Processing Center.
VISA_CLASS	Indicates the type of temporary application submitted for processing. R = H-1B; A = E-3 Australian; C = H-1B1 Chile; S = H-1B1 Singapore. Also referred to as “Program” in prior years.
EMPLOYMENT_START_DATE	Beginning date of employment.
EMPLOYMENT_END_DATE	Ending date of employment.
EMPLOYER_NAME	Name of employer submitting labor condition application.
EMPLOYER_BUSINESS_DBA	Trade Name or dba name of employer submitting labor condition application, if applicable.
EMPLOYER_ADDRESS	Contact information of the Employer requesting temporary labor certification.
EMPLOYER_CITY	
EMPLOYER_STATE	
EMPLOYER_POSTAL_CODE	
EMPLOYER_COUNTRY	
EMPLOYER_PROVINCE	
EMPLOYER_PHONE	
EMPLOYER_PHONE_EXT	
AGENT_REPRESENTING_EMPLOYER	Y = Employer is represented by an Agent or Attorney; N = Employer is not represented by an Agent or Attorney.
AGENT_ATTORNEY_NAME	Name of Agent or Attorney filing an H-1B application on behalf of the employer.

FIELD NAME	DESCRIPTION
AGENT_ATTORNEY_CITY	City information for the Agent or Attorney filing an H-1B application on behalf of the employer.
AGENT_ATTORNEY_STATE	State information for the Agent or Attorney filing an H-1B application on behalf of the employer.
JOB_TITLE	Title of the job.
SOC_CODE	Occupational code associated with the job being requested for temporary labor condition, as classified by the Standard Occupational Classification (SOC) System.
SOC_NAME	Occupational name associated with the SOC_CODE.
NAICS_CODE	Industry code associated with the employer requesting permanent labor condition, as classified by the North American Industrial Classification System (NAICS).
TOTAL_WORKERS	Total number of foreign workers requested by the Employer(s).
NEW_EMPLOYMENT	Indicates requested worker(s) will begin employment for new employer, as defined by USCIS I-29.
CONTINUED_EMPLOYMENT	Indicates requested worker(s) will be continuing employment with same employer, as defined by USCIS I-29.
CHANGE_PREVIOUS_EMPLOYMENT	Indicates requested worker(s) will be continuing employment with same employer without material change to job duties, as defined by USCIS I-29.
NEW_CONCURRENT_EMPLOYMENT	Indicates requested worker(s) will begin employment with additional employer, as defined by USCIS I-29.
CHANGE_EMPLOYER	Indicates requested worker(s) will begin employment for new employer, using the same classification currently held, as defined by USCIS I-29.
AMENDED_PETITION	Indicates requested worker(s) will be continuing employment with same employer with material change to job duties, as defined by USCIS I-29.
FULL_TIME_POSITION	Y = Full Time Position; N = Part Time Position.
PREVAILING_WAGE	Prevailing Wage for the job being requested for temporary labor condition.
PW_UNIT_OF_PAY	Unit of Pay. Valid values include “Daily (DAI),” “Hourly (HR),” “Bi-weekly (BI),” “Weekly (WK),” “Monthly (MTH),” and “Yearly (YR)”.
PW_WAGE_LEVEL	Variables include "I", "II", "III", "IV" or "N/A."
PW_SOURCE	Variables include "OES", "CBA", "DBA", "SCA" or "Other".
PW_SOURCE_YEAR	Year the Prevailing Wage Source was Issued.
PW_SOURCE_OTHER	If "Other Wage Source", provide the source of wage.
WAGE_RATE_OF_PAY_FROM	Employer’s proposed wage rate.
WAGE_RATE_OF_PAY_TO	Maximum proposed wage rate.
WAGE_UNIT_OF_PAY	Unit of pay. Valid values include “Hour", "Week", "Bi-Weekly", "Month", or "Year".
H-1B_DEPENDENT	Y = Employer is H-1B Dependent; N = Employer is not H-1B Dependent.
WILLFUL_VIOLATOR	Y = Employer has been previously found to be a Willful Violator; N = Employer has not been considered a Willful Violator.
SUPPORT_H1B	Y = Employer will use the temporary labor condition application only to support H-1B petitions or extensions of status of exempt H-1B worker(s); N = Employer will not use the temporary labor condition application to support H-1B petitions or extensions of status for exempt H-1B worker(s);
LABOR_CON_AGREE	Y = Employer agrees to the responses to the Labor Condition Statements as in the subsection; N = Employer does not agree to the responses to the Labor Conditions Statements in the subsection.
PUBLIC_DISCLOSURE_LOCATION	Variables include "Place of Business" or "Place of Employment."
WORKSITE_CITY	City information of the foreign worker's intended area of employment.
WORKSITE_COUNTY	County information of the foreign worker's intended area of employment.
WORKSITE_STATE	State information of the foreign worker's intended area of employment.
WORKSITE_POSTAL_CODE	Zip Code information of the foreign worker's intended area of employment.
ORIGINAL_CERT_DATE	Original Certification Date for a Certified_Withdrawn application.

2.3 Data Cleaning

The raw merged data is messy and some cleaning steps needs to be performed on the data table before any exploratory data analysis can be performed on it. These are the data cleaning steps that are performed to get our dataset into the right shape:

a. Removing all the N/A values and only considering complete cases.

b. The merged dataset has filing records for other kinds of visa processing such as H1-B1, H2-B2 etc. So, these records are filtered out to retain only those records corresponding to H-1B data.

c. Normalizing the response variable “CASE_STATUS” values to only two levels: APPROVED or NOT APPROVED.

d. Correcting class imbalance in the response variable. Making the number of approved and Not approved case status same in the data, in order to remove the class imbalance. To take care of the class imbalance we used oversampling and under sampling both.

e. Reduced the file size to 81MB after only taking complete cases. The total number of complete observations are 624,538. There are only 22 variables considered as these are of interest for our analysis. Out of which 8 are categorical variables and 14 are numeric variables.

f. We have to build a common scale for salary data for part time and full time em ployees so that it becomes easy to compare them.We do that by scaling the hourly, Weekly, Bi-Weekly and monthly salaries to an yearly pay.

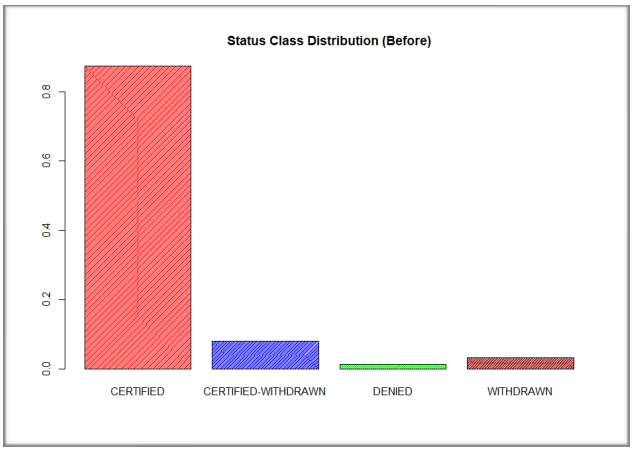
g. We created new variables based on the different date variables:

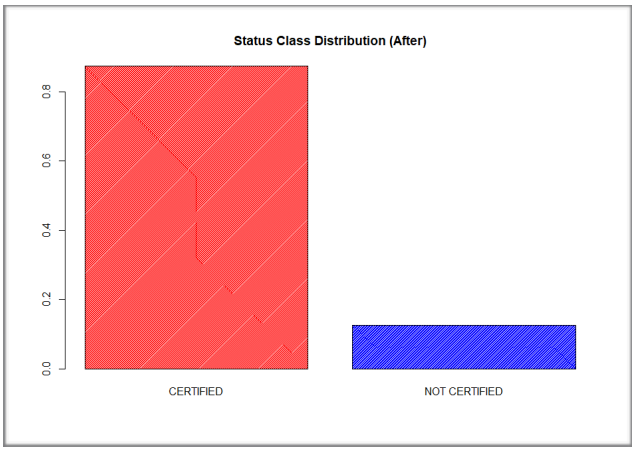
   1) SUBMIT_DECISION= DECISION_DATE-CASE_SUBMIT
   2) START_END=CASE_EMPLOYMENT_START_DATE-CASE_EMPLOY MENT_END_DATE
   3) START_DECISION=CASE_EMPLOYMENT_START_DATE-DECISION_ DATE)
   4) END_DECISION=CASE_EMPLOYMENT_END_DATE-DECISION_ DATE)
   5) SUBMIT_END=CASE_SUBMIT-CASE_EMPLOYMENT_END_DATE
   6) SUBMIT_START=CASE_SUBMIT-CASE_EMPLOYMENT_START_DATE

h. Created a new variable ‘REGION’ which normalized and divided the 55 states into only 4 different regions (4 levels) for Employer Location and Work Location region.

2.4 Exploratory Analysis

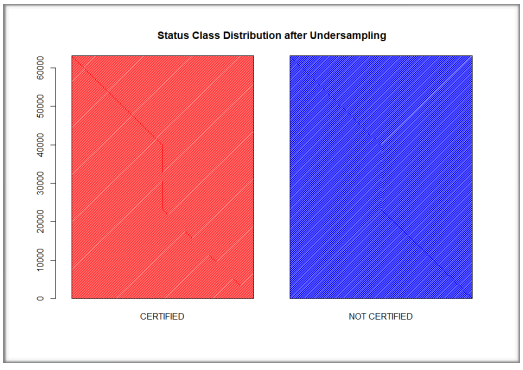
2.4.1 Analyzing the Response variable before and after normalizing





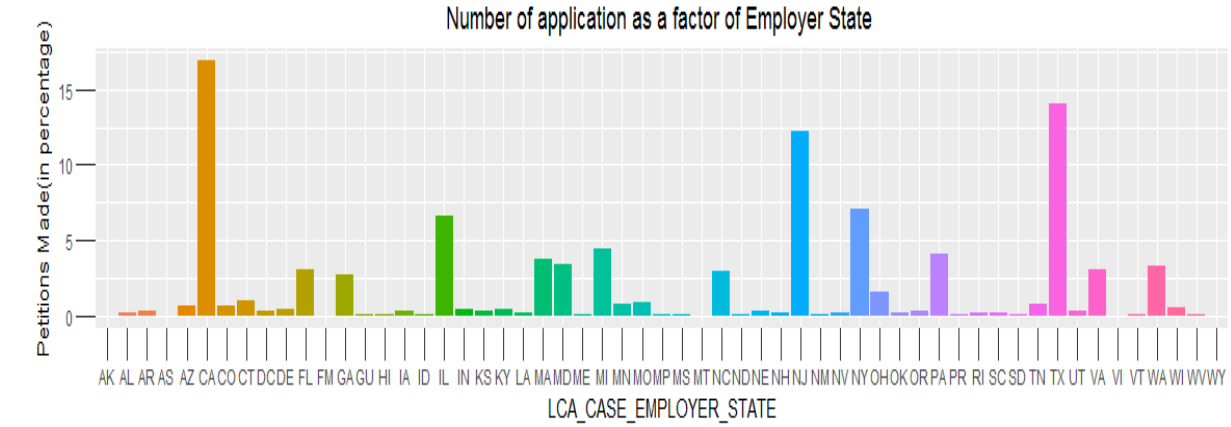
We can see that there is a class imbalance even after normalizing the values of the response variable as just APPROVED and NOT APPROVED. Due to this imbalance the models that we would fit on our data would almost alway predict the response as APPROVED.

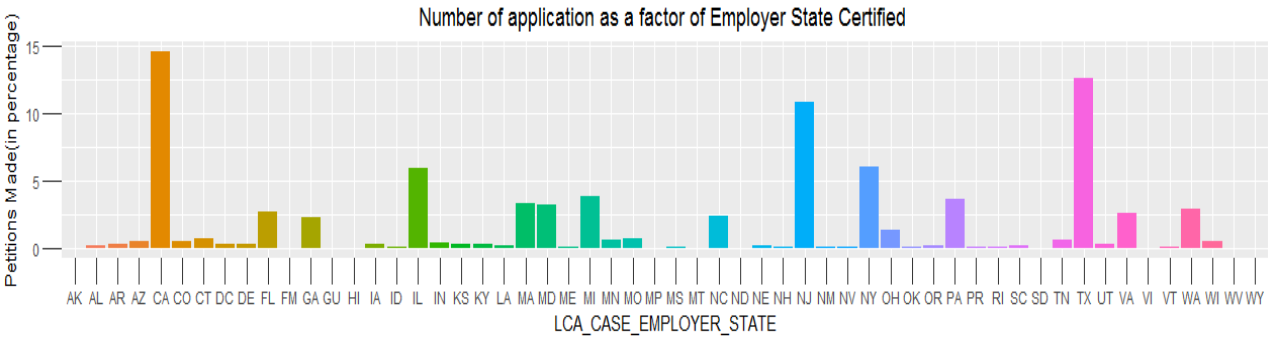
2.4.2 Resolving Class Imbalance Issue:



We used the R Package 'ROSE' - Random Oversampling Examples, in which we tried under sampling, oversampling and both together and then fitting models: Logistic Regression and K Nearest Neighbors.

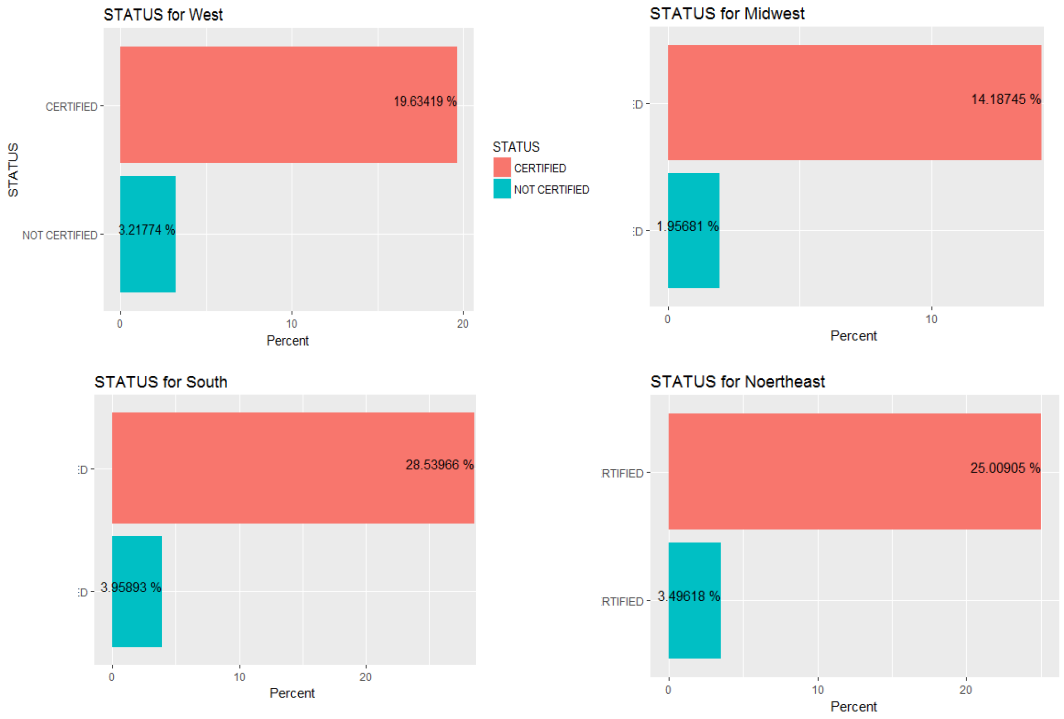
2.4.3. Number of H-1B submissions by State





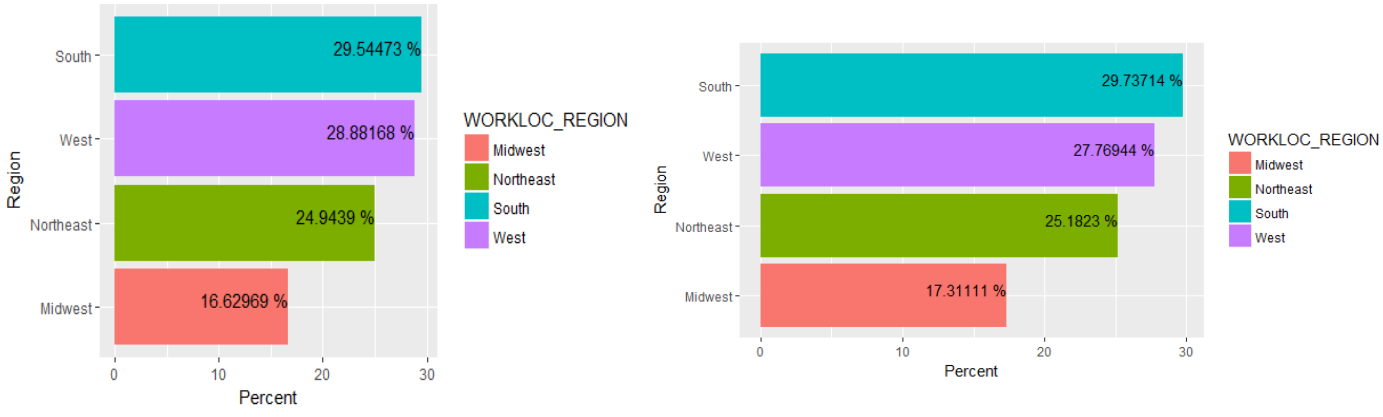
This again shows the class imbalance present in our data. We can see that the graph for total number of H-1B submissions and the Approved H-1B applications are almost the same across different states. The top states include California, Illinois, New Jersey and Texas.

2.4.4 Case Status based on different regions



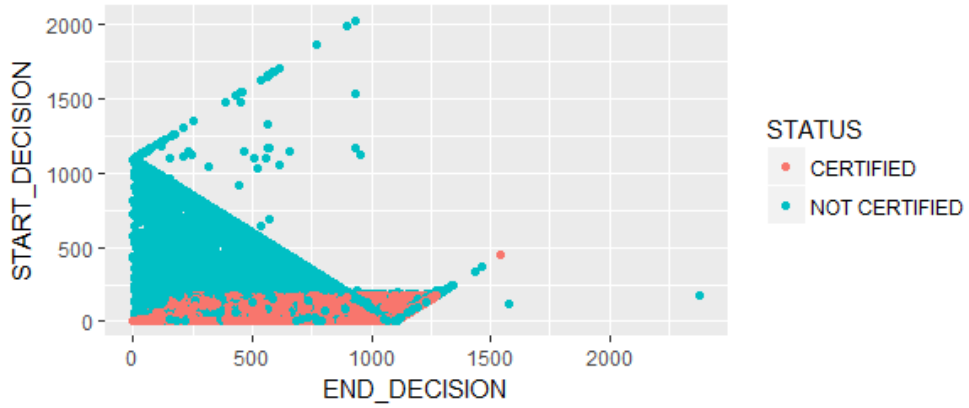
Based on the state information we created a new region variable which divides the different states into four regions: West, Midwest, South and Northeast. The class imbalance is again visible through these graphs

2.4.5 Case Status based on Work Location Region



Just like regions we divided the Work Location into different regions based on which state the region falls into. The left shows Approved Case status and the right graph shows the Not Approved case status for the different Work Location regions.

2.4.6 Case Status based on Decision Dates



The decision dates made the fitted models very predictable which was not realistic and also these columns will not be available in future prediction data as one will not know the decision end date beforehand. So, we decided to remove these variables and then fitting the data.

2.5 Analysis

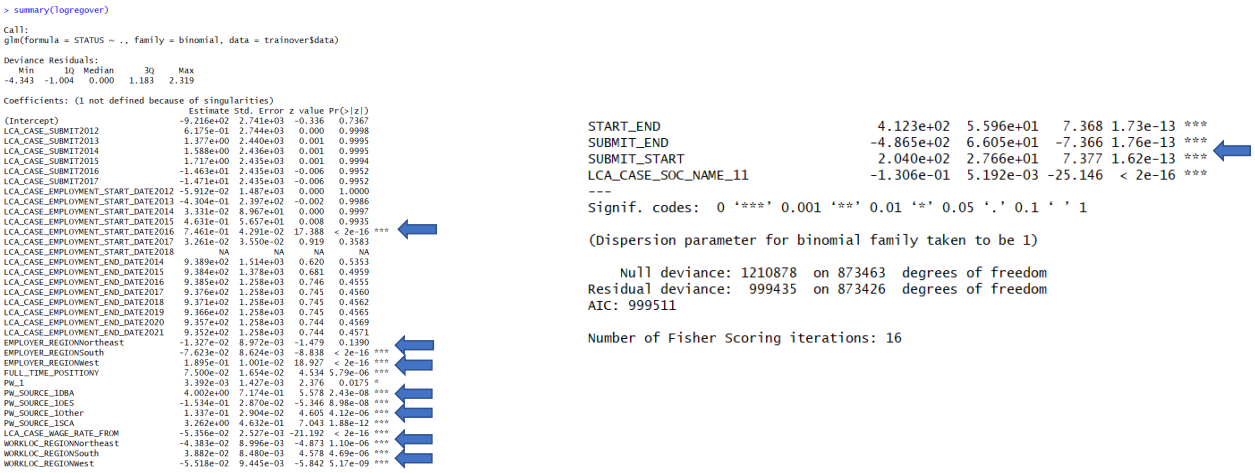
Approach: We tried fitting our data using different models which could take care of categorical and numerical variables. The data was split into two parts: 70% training data on which we fit the model and 30% testing data on which we make predictions using our model and calculate the accuracy.

2.5.1 Logistic Regression

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

The goal of logistic regression is to find the best fitting (yet biologically reasonable) model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression generates the coefficients (and its standard errors and significance levels) of a formula to predict a logit transformation of the probability of presence of the characteristic of interest.

Before performing Logistic Regression, the class imbalance in the data was removed. First analyzed the summary of the model to look for significant variables in the model:



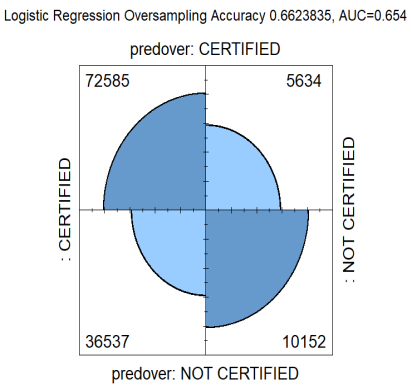
```
> summary(logregover)

Call:
glm(formula = STATUS ~ ., family = binomial, data = trainover$data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.343  -1.004   0.000   1.183   2.319

Coefficients: (1 not defined because of singularities)
                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -9.216e+02  2.741e+03  -0.336   0.7367
LCA_CASE_SUBMIT2012              6.175e-01  2.744e+03   0.000   0.9998
LCA_CASE_SUBMIT2013              1.377e+00  2.440e+03   0.001   0.9995
LCA_CASE_SUBMIT2014              1.588e+00  2.436e+03   0.001   0.9995
LCA_CASE_SUBMIT2015              1.717e+00  2.435e+03   0.001   0.9994
LCA_CASE_SUBMIT2016             -1.463e+01  2.435e+03  -0.006   0.9952
LCA_CASE_SUBMIT2017             -1.471e+01  2.435e+03  -0.006   0.9952
LCA_CASE_EMPLOYMENT_START_DATE2012 -5.912e-02  1.487e+03   0.000   1.0000
LCA_CASE_EMPLOYMENT_START_DATE2013 -4.304e-01  2.397e+02  -0.002   0.9986
LCA_CASE_EMPLOYMENT_START_DATE2014  3.331e-02  8.967e+01   0.000   0.9997
LCA_CASE_EMPLOYMENT_START_DATE2015  4.631e-01  5.657e+01   0.008   0.9935
LCA_CASE_EMPLOYMENT_START_DATE2016  7.461e-01  4.291e-02  17.388  < 2e-16 ***
LCA_CASE_EMPLOYMENT_START_DATE2017  3.261e-02  3.550e-02   0.919   0.3583
LCA_CASE_EMPLOYMENT_START_DATE2018         NA         NA      NA       NA
LCA_CASE_EMPLOYMENT_END_DATE2014    9.389e+02  1.514e+03   0.620   0.5353
LCA_CASE_EMPLOYMENT_END_DATE2015    9.384e+02  1.378e+03   0.681   0.4959
LCA_CASE_EMPLOYMENT_END_DATE2016    9.385e+02  1.258e+03   0.746   0.4555
LCA_CASE_EMPLOYMENT_END_DATE2017    9.376e+02  1.258e+03   0.745   0.4560
LCA_CASE_EMPLOYMENT_END_DATE2018    9.371e+02  1.258e+03   0.745   0.4562
LCA_CASE_EMPLOYMENT_END_DATE2019    9.366e+02  1.258e+03   0.745   0.4565
LCA_CASE_EMPLOYMENT_END_DATE2020    9.357e+02  1.258e+03   0.744   0.4569
LCA_CASE_EMPLOYMENT_END_DATE2021    9.352e+02  1.258e+03   0.744   0.4571
EMPLOYER_REGIONNortheast        -1.327e-02  8.972e-03  -1.479   0.1390
EMPLOYER_REGIONSouth            -7.623e-02  8.624e-03  -8.838  < 2e-16 ***
EMPLOYER_REGIONWest              1.895e-01  1.001e-02  18.927  < 2e-16 ***
FULL_TIME_POSITIONY              7.500e-02  1.654e-02   4.534 5.79e-06 ***
PW_1                             3.392e-03  1.427e-03   2.376   0.0175 *
PW_SOURCE_1DBA                   4.002e+00  7.174e-01   5.578 2.43e-08 ***
PW_SOURCE_1OES                  -1.534e-01  2.870e-02  -5.346 8.98e-08 ***
PW_SOURCE_1Other                 1.337e-01  2.904e-02   4.605 4.12e-06 ***
PW_SOURCE_1SCA                   3.262e+00  4.632e-01   7.043 1.88e-12 ***
LCA_CASE_WAGE_RATE_FROM         -5.356e-02  2.527e-03 -21.192  < 2e-16 ***
WORKLOC_REGIONNortheast         -4.383e-02  8.996e-03  -4.873 1.10e-06 ***
WORKLOC_REGIONSouth              3.882e-02  8.480e-03   4.578 4.69e-06 ***
WORKLOC_REGIONWest              -5.518e-02  9.445e-03  -5.842 5.17e-09 ***
START_END                        4.123e+02  5.596e+01   7.368 1.73e-13 ***
SUBMIT_END                      -4.865e+02  6.605e+01  -7.366 1.76e-13 ***
SUBMIT_START                     2.040e+02  2.766e+01   7.377 1.62e-13 ***
LCA_CASE_SOC_NAME_11            -1.306e-01  5.192e-03 -25.146  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1210878  on 873463  degrees of freedom
Residual deviance:  999435  on 873426  degrees of freedom
AIC: 999511

Number of Fisher Scoring iterations: 16
```

The arrows and asterisk indicate that the variable is significant in the model. We dropped these insignificant variables one by one and then fit the Logistic Regression model again. After which we used this model for prediction on our test data.

Results:

The confusion matrix for the test data for different sampling scenarios:

1. Undersampling



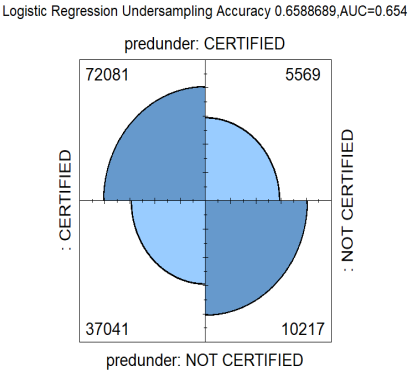
The accuracy for this case is 66.24%

Sensitivity = 92.78%

Specificity = 21.74%

AUC = 65.4%

2. Oversampling



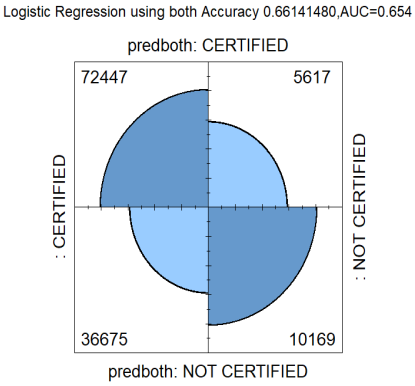
The accuracy for this case is 65.89%

Sensitivity = 92.83%

Specificity = 21.62%

AUC = 65.4%

3. Both



The accuracy for this case is 66.14%

Sensitivity = 92.80%

Specificity = 21.71%

AUC = 65.4%

2.5.2 K-Nearest Neighbors

KNN is one of the most basic methods when going for classification. Computing time for KNN is highest among all the other methods for this dataset. KNN requires all data to be in numeric form because it takes an array of vectors only. Hence all categorical attributes were transformed to one-hot encoded variables & numeric attributes were normalized. Since it is said that the best value of k would be $\sqrt{n}$ where n is the number of data points, we tried taking k=800. However that was too computationally expensive in terms of time.

Tried different values of k ranging from 1 to 499. Best accuracy obtained was for k= 20. We were unable to perform cross validation since the computational time for this was extremely high considering the size of this data set. KNN also suffers from the curse of

dimensionality; hence we implemented other methods. The results of different cases and values of K are mentioned in the Appendix.

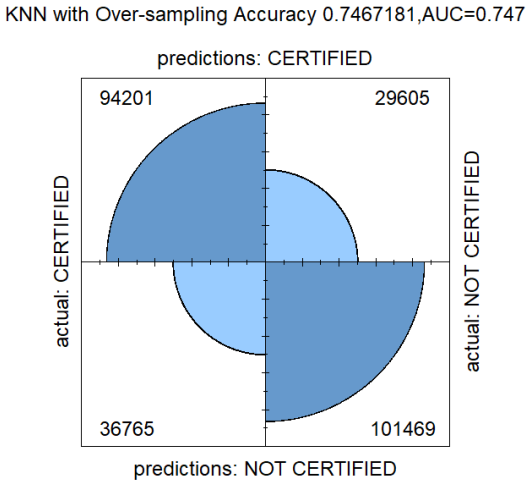
Result:

For K = 20 we got the best accuracy

	Under sampled Dataset	Over sampled Dataset	Original Dataset
Test Accuracy :	68.75%	74.67%	91.33%

Predicted	Actual		
		CERTIFIED	NOT CERTIFIED
	CERTIFIED	162932	15599
	NOT CERTIFIED	636	8195

The oversampled data gave the best and more realistic accuracy:



The accuracy for this case is 74.67%

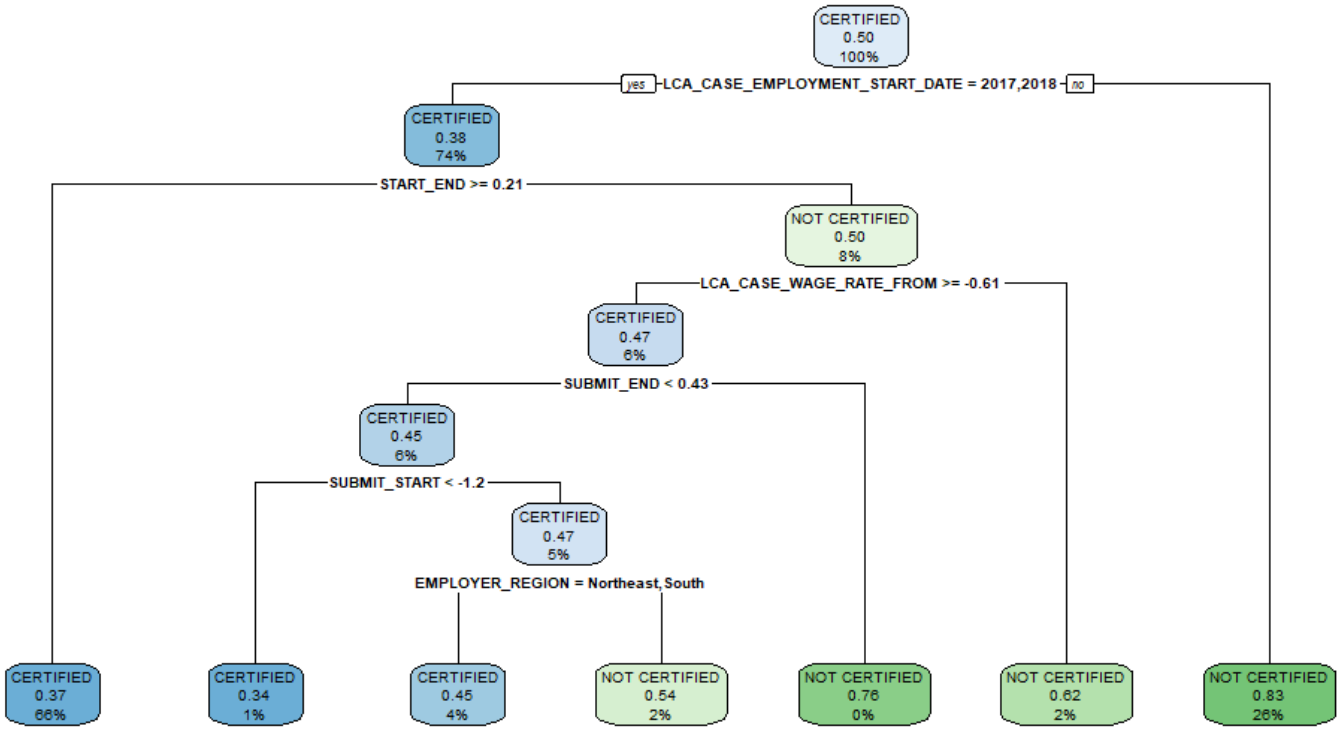
Sensitivity = 76.09%

Specificity = 73.40%

AUC = 74.7%

2.5.3 Recursive Partitioning and Regression Trees (RPART)

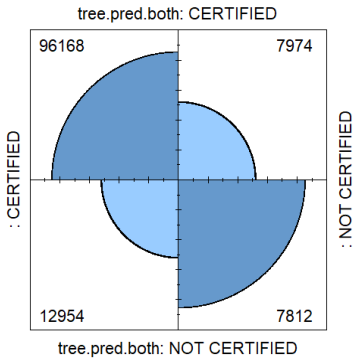
The rpart algorithm works by splitting the dataset recursively, which means that the subsets that arise from a split are further split until a predetermined termination criterion is reached. At each step, the split is made based on the independent variable that results in the largest possible reduction in heterogeneity of the dependent (predicted) variable. It creates binary decision trees and decision rules that has more sensitivity and specificity.



Result:

The confusion matrix is shown below:

Recursive Partitioning and Regression Trees using both Accuracy 0.8324527,AUC=0.688



The accuracy for this case is 83.25%

Sensitivity = 92.34%

Specificity = 37.62%

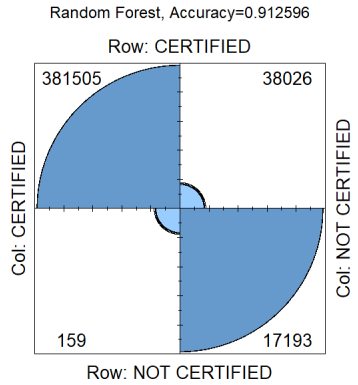
AUC = 68.8%

2.5.4 RandomForest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

Result:

The confusion matrix is shown below:



The accuracy for this case is 91.25%

Sensitivity = 90.94%

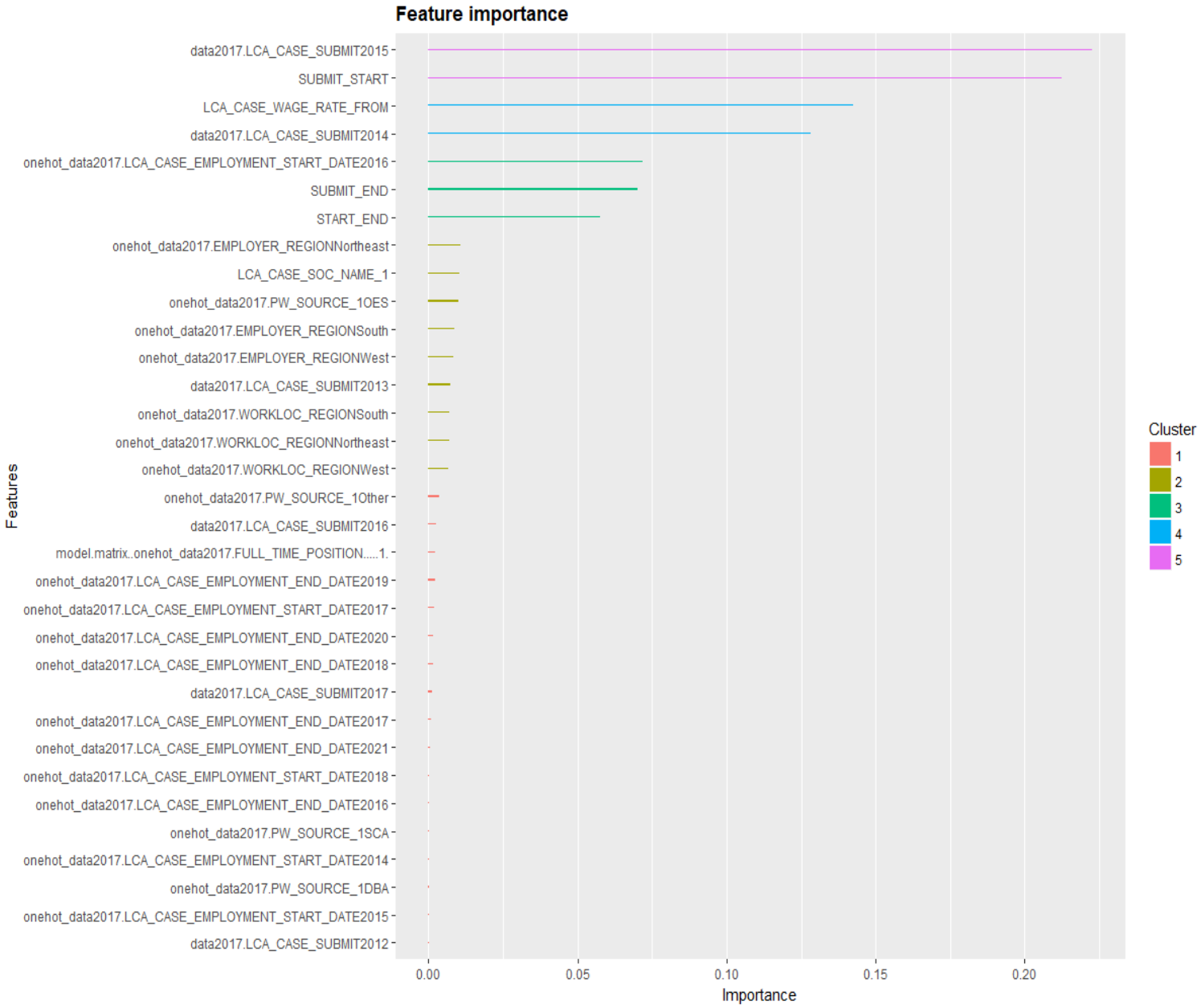
Specificity = 99.08%

2.5.5 Extreme Gradient Boosting (XGBoost)

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solved our problem in a fast and accurate way. It iteratively combines a number of “weak” learners to create a strong classifier. The weak learner here is a decision tree. The learners are added based on a loss function which is to be minimized at each iteration. It regularizes model formalization to control over-fitting. It took two seconds to build a model on the training data. The parameters chosen were:

a. Max depth = 15
b. Eta = 0.5
c. nrounds = 19

We first tried to analyze the importance of the variables in our model with the importance() function in R.

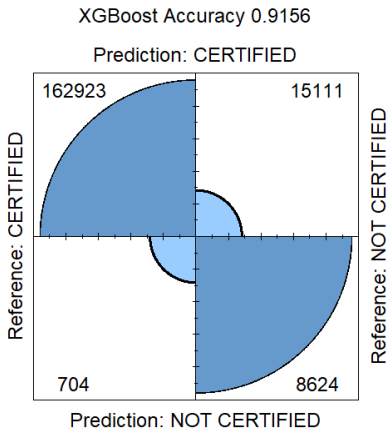


The top 3 important features in our model are LCA_CASE_SUBMIT, SUBMIT_START and LCA_CASE_WAGE_RATE_FROM. This ideally also makes sense because the H-1B visa allocation depends a lot upon when the application was submitted and also on the

salary slab that the applicant falls into. The higher the salary, the better the chance of getting approved.

Result:

The confusion matrix is shown below:



The accuracy for this case is 91.56%

Sensitivity = 91.51%

Specificity = 92.45%

3. Discussion & Conclusion

We tried fitting our H-1B training dataset using 5 different machine learning algorithm and a summary for them is in the table below:

Training Method	Test Accuracy
Logistic Regression (Undersampling, Oversampling, both)	It is almost 66% for all the sampling techniques
K-Nearest Neighbors (K = 20)	74.67%
RPART	83.25%
RandomForest	91.25%
XGBoost	91.56%

XGBoost gives us the best testing accuracy in comparison to all the methods. RandomForest also gives us the same amount of accuracy but it is computationally longer in comparison to XGBoost and thus, we conclude that we achieved the best testing accuracy for our data with XGBoost, which is 91.56%. Based on this we can say that

given a known set of variables the H1-B case status can be predicted. Using the importance function we found that the most significant variables in our model are LCA_CASE_SUBMIT, SUBMIT_START and LCA_CASE_WAGE_RATE_FROM followed by REGION and whether the applicant has a FULL TIME POSITION or not. Even in reality we know that factors like Wage, when the application was submitted, region and whether the applicant has a full time position or not has a big influence on the decision.

Future Scope

The project is only based on the FY 2017 data but we have data available from 2011-2016 as well. We would try to levy the FY 2017 data model on these datasets and analyze the changes in the visa allocation process throughout these years. Moreover see if we can predict with a good accuracy on these datasets as well.

The H-1B visa application and process is very expensive, which is why it is really important for an employer to know how good are the chances of the employee, filing for the visa, getting approved. This is why we aim to build a recommender system for companies and employers which will help them in deferring whether the employee's H-1B visa will be Approved or Not Approved.

4. References

1. https://eight2late.wordpress.com/2016/02/16/a-gentle-introduction-to-decision-trees-using-r/
2. https://www.foreignlaborcert.doleta.gov
3. https://github.com/dmlc/xgboost
4. https://www.kaggle.com/nsharan/h-1b-visa
5. https://www.medcalc.org/manual/logistic_regression.php
6. https://en.wikipedia.org/wiki/Random_forest

5. Appendices

1. Results for different values K and sampling cases

No sampling k=20

```
> table(pred_knn,test$STATUS)

pred_knn       CERTIFIED NOT CERTIFIED
  CERTIFIED       162932         15599
  NOT CERTIFIED      636          8195
> mean(pred_knn==test$STATUS)
[1] 0.9133496
```

For under sampling k= 101

```
pred_knn_under  CERTIFIED NOT CERTIFIED
  CERTIFIED         26740         15175
  NOT CERTIFIED      4726         16463
> mean(pred_knn_under==test$STATUS)
[1] 0.6846317
```

For under sampling k= 201

```
pred_knn_under  CERTIFIED NOT CERTIFIED
  CERTIFIED         27422         15946
  NOT CERTIFIED      4044         15692
> mean(pred_knn_under==test$STATUS)
[1] 0.6832213
```

For under sampling k= 51

```
pred_knn_under  CERTIFIED NOT CERTIFIED
  CERTIFIED         26170         14378
  NOT CERTIFIED      5296         17260
> mean(pred_knn_under==test$STATUS)
[1] 0.688229
```

For under sampling k= 40

```
pred_knn_under  CERTIFIED NOT CERTIFIED
  CERTIFIED         26058         14261
  NOT CERTIFIED      5408         17377
> mean(pred_knn_under==test$STATUS)
[1] 0.6883082
```

For under sampling k= 20

```
pred_knn_under  CERTIFIED NOT CERTIFIED
  CERTIFIED         25101         13366
  NOT CERTIFIED      6365         18272
```

```
> mean(pred_knn_under==test$STATUS)
[1] 0.6873257
```

K=40 full dataset

```
pred_knn        CERTIFIED NOT CERTIFIED
  CERTIFIED        163103         16016
  NOT CERTIFIED       465          7778
> mean(pred_knn==test$STATUS)
[1] 0.9120366
```

K=20 over sampling

```
pred_knn_above  CERTIFIED NOT CERTIFIED
  CERTIFIED         94201         29605
  NOT CERTIFIED     36765        101469
> mean(pred_knn_above==test$STATUS)
[1] 0.7467181
```