SAN DIEGO STATE UNIVERSITY

# MIS 620 Group Project Report

Team: American Dream

Jake Bellamy, Yadong Lu, Jay Sun, Preethi Narayanan

Instructor: Dr. Aaron Elkins

Table of Contents

## Executive Summary

In 2016, there are 1,183,505 persons obtaining lawful permanent resident status, among those 137,893 persons are on employment-based preferences (from homeland security department). Getting a working visa and then applying for green card is becoming a main path for people outside of United States to immigrating to this country and fulfill their "American Dream". As a team consists of immigrants or descendants come from immigrant families, we are interested in digging valuable information from employment-based visa data. After researching, we agreed on analysing H1B Disclosure Data, not only because H1B is the most common visa status applied for and held by foreign workers, but also because its data is easy for public to access and perform.

We use the 2018 H-1B application disclosure data to analyze petitions geographical distribution, job positions and employers with the most applications, which type or condition of applications are more likely to be denied or withdrawn. With the discoveries of our research, candidates who seek to work under H1B visa and US employers who seek to hire H1B workers can improve approved rate of their petitions.

Though the dataset we use comes with easily accessible formats, it still need some further cleaning before it could be used to analysis.Cleaning process includes: remove irrelevant attributes, uniform measuring units, balance the data.

As the data ready for analyzing, we came with following hypotheses:

1.The mean of wage offered is same for candidates whose visa has been approved is same as those candidates whose visa has been denied.

2. Employer_name has a significant relationship in case being approved or rejected.

3. The case status has no relationship with job title of the candidate.

Our top 3 goals are:

1. Determine factor that are important to classify the decision.
2. Build models to predict the decision of an application.
3. Interpret the result of the models and provide insight that would allow individuals to obtain their H1B.

We built and tested following prediction models: GAM, GLMNET, Boosted Logistic Regression, Decision Tree, Naive Bayes, Linear Discriminant.

After running all the models we found GLM performing the best on accuracy, while Naive Bayes performing the worst. Among the important predictors, SOC_NAME, WILLFUL_VIOLATOR, FULL_TIME_POSITIONS received more percentage than the others. We can say that a H1b application with employee as a full-time computer occupation practitioner, submitted by a non-willful violator US employer has the highest rate to be certified.

On April 18, 2017, President Trump signed the “Buy American and Hire American Executive Order” , which has brought uncertainty to the process of H1B. In the future, we would do more research to compare the disclosure data before and after the executive order, to see what differences it leads.

## Discovery

The H-1B is an employment-based, non-immigrant visa category for temporary foreign workers in the United States. For a foreign candidate applying for H-1B visa, an US employer must offer a job and submit a petition for H-1B visa to the US immigration department. The US immigration department (USCIS) selects 85,000 applications through a random process from over 200,000 petitions each year. The application data is available at OFLC Case Disclosure Data page, allowing the public to access and perform in-depth longitudinal research and analysis. This data provides key insights into H-1B visa cases, which bring valuable information to H-1B applicants and their US employers.

Based on our research goals, we choose a dataset from Kaggle called "1. Master H1B Dataset" which was downloaded and pre-cleaned by a forerunner. The reasons that we choose are: first, the dataset is rich enough for us to analysis and achieve our top research goals, second, it is also vary clean which could save our time from basic cleaning and give us more time to build and test our models.

The data comes in CSV file, contains 528,101 rows and 27 columns.

| Variable | Type | Description |
|---|---|---|
| CASE_SUBMITTED_DAY | Integer | Date the application was submitted. |
| CASE_SUBMITTED_MONTH | Integer | Month the application was submitted. |
| CASE_SUBMITTED_YEAR | Integer | Year the application was submitted. |
| DECISION_DAY | Integer | Day on which the last significant event or decision |

| | | |
|---|---|---|
| | | was recorded by the Chicago National Processing Center. |
| DECISION_MONTH | Integer | Month on which the last significant event or decision was recorded by the Chicago National Processing Center. |
| DECISION_YEAR | Integer | Year on which the last significant event or decision was recorded by the Chicago National Processing Center. |
| VISA_CLASS | Factor with 4 levels | Indicates the type of temporary application submitted for processing. |
| EMPLOYER_NAME | Factor with 61104 levels | Name of employer submitting labor condition application. |
| EMPLOYER_STATE | Factor with 57 levels | State of employer submitting labor condition application. |
| EMPLOYER_COUNTRY | Factor with 4 levels | Country of employer submitting labor condition application. |
| SOC_NAME | Factor with 56 levels | Occupational name associated with the SOC_CODE. |
| NAICS_CODE | Integer | Industry code associated with the employer requesting permanent labor condition, as classified by the North American Industrial Classification System (NAICS) . |
| TOTAL_WORKERS | Integer | Total number of foreign workers requested by the Employer. |
| FULL_TIME_POSITION | Factor with 3 levels | Y = Full Time Position; N = Part Time Position |

| | | |
|---|---|---|
| PREVAILING_WAGE | Number | Prevailing Wage for the job being requested for temporary labor condition. |
| PW_UNIT_OF_PAY | Factor with 6 levels | Unit of Pay. Variables include “YEAR”, “MONTH”, “BI-WEEKLY”, “WEEKLY”, “HOUR” |
| PW_SOURCE | Factor with 6 levels | Variables include "OES", "CBA", "DBA", "SCA" or "Other". |
| PW_SOURCE_YEAR | Integer | Year the Prevailing Wage Source was Issued. |
| PW_SOURCE_OTHER | Factor with 236 levels | Other Wage Source provide the source of Prevailing wage |
| WAGE_RATE_OF_PAY_FROM | Number | Employer’s proposed wage rate. |
| WAGE_RATE_OF_PAY_TO | Number | Maximum proposed wage rate. |
| WAGE_UNIT_OF_PAY | Factor with 6 levels | Unit of pay. |
| H-1B_DEPENDENT | Factor with 3 levels | Y = Employer is H-1B Dependent; N = Employer is not H-1B Dependent. |
| WILLFUL_VIOLATOR | Factor with 3 levels | Y = Employer has been previously found to be a Willful Violator; N = Employer has not been considered a Willful Violator |
| WORKSITE_STATE | Factor with 55 levels | State information of the foreign worker's intended area of employment. |
| WORKSITE_POSTAL_CODE | Factor with 12178 levels | Zip Code information of the foreign worker's intended area of employment. |
| CASE_STATUS | Factor with 4 levels | 4 Status associated with the |

| | | |
|---|---|---|
| | | last significant event or decision. Valid values include “Certified,” “Certified-Withdrawn,” Denied,” and “Withdrawn”. |

## Data Preparation

Though the dataset we use comes with easily accessible formats, it still need some further cleaning before it could be used to analysis.Cleaning process includes but not limits to:

1. Remove irrelevant attributes
2. Uniform measuring units
3. Balance the data

Remove irrelevant attributes:

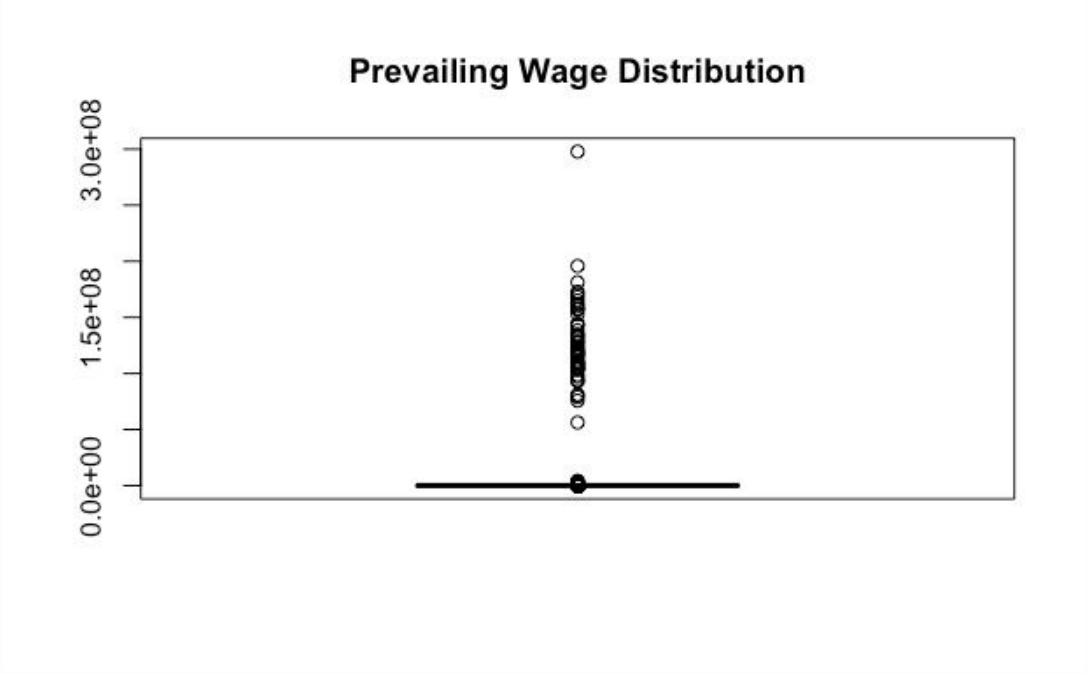
There are total 27 variables in the dataset, some of them do nothing with our analysis and prediction, or would cause our models crushed. We removed those variables for reasons:

| Variables | Reason to be removed | Way of removing |
|---|---|---|
| CASE_SUBMITTED_DAY<br>DECISION_DAY | Rather than YEAR or MONTH data, which can contrite trend analysis, DAYs are more likely arbitrary values | Remove directly |
| VISA_CLASS | Since we focus on H1B visa, other class visa are not in the extent of our analysis | First remove rows with VISA_CLASS other than H1B, then remove the variable |
| EMPLOYER_COUNTRY | We focus on American employers | First remove rows with EMPLOYER_COUNTRY other than United States, then remove the variable |
| NAICS_CODE | We take positions of employees rather than industry of employers | Remove directly |
| TOTAL_WORKERS | We focus on individual H1B cases | First remove rows with TOTAL_WORKERS other than “1”, then remove the variable |

| | | |
|---|---|---|
| PW_SOURCE<br>PW_SOURCE_YEAR<br>PW_SOURCE_OTHER | These three variables do nothing with our research | Remove directly |
| WAGE_RATE_OF_PAY_FROM<br>WAGE_RATE_OF_PAY_TO<br>WAGE_UNIT_OF_PAY | We take Prevailing Wage rather than employers proposed wage rate | Remove directly |
| WORKSITE_POSTAL_CODE | This variable has too much dimension which can cause model crashing | Remove directly |

Uniform measuring units：

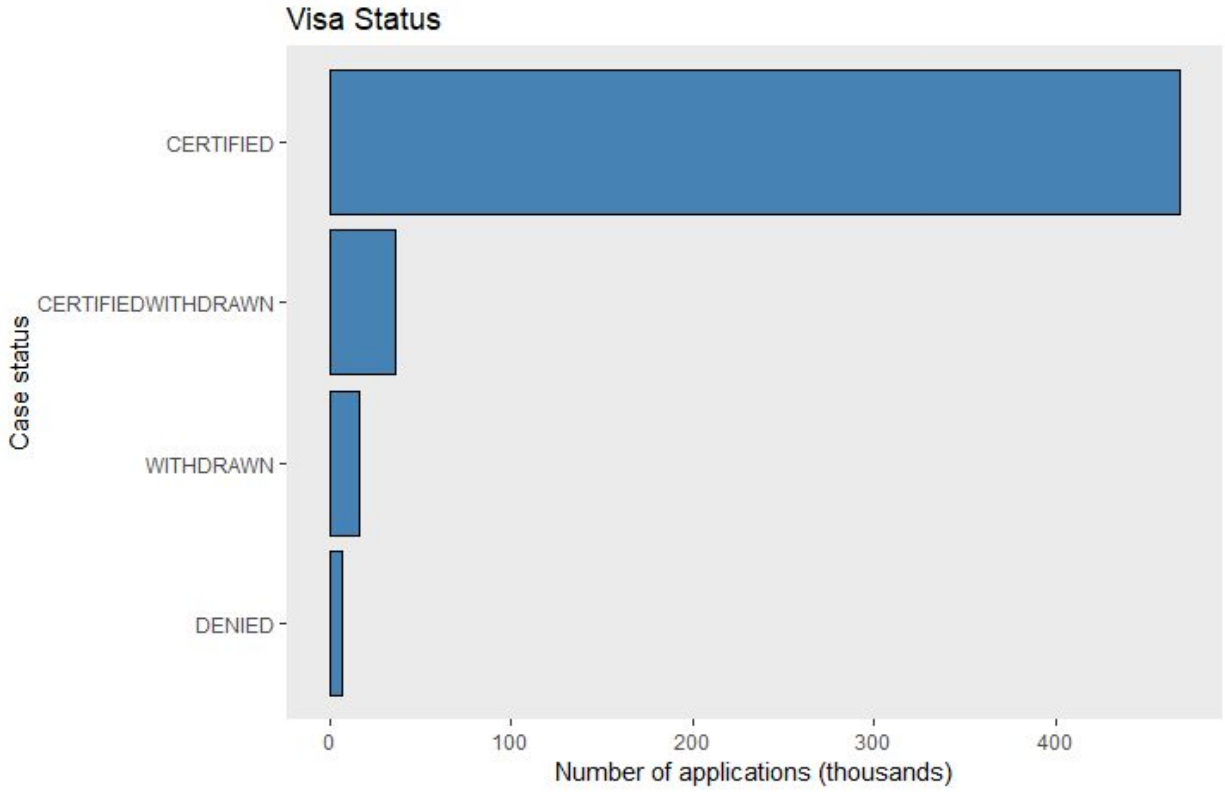
To use wages as a predictor of our models, we first need to convert them to the same time scale of payment. In our dataset, the following values of payment rate exist. While 93% of the records provide Wage at the Year scale, we convert Wages not in Year scale to the Year scale. Then, we remove the variable column “WAGE_UNIT_OF_PAY”. After converting the payment unit, we saw unreasonable value in our “prevailing_wage”, with minimum “0” and maximum “297785280” (See “Prevailing Wage Distribution”). These may came from typos of original data or the converting process or both. We used “boxplot” function to find a reasonable range of wages by omitting significant outliers.

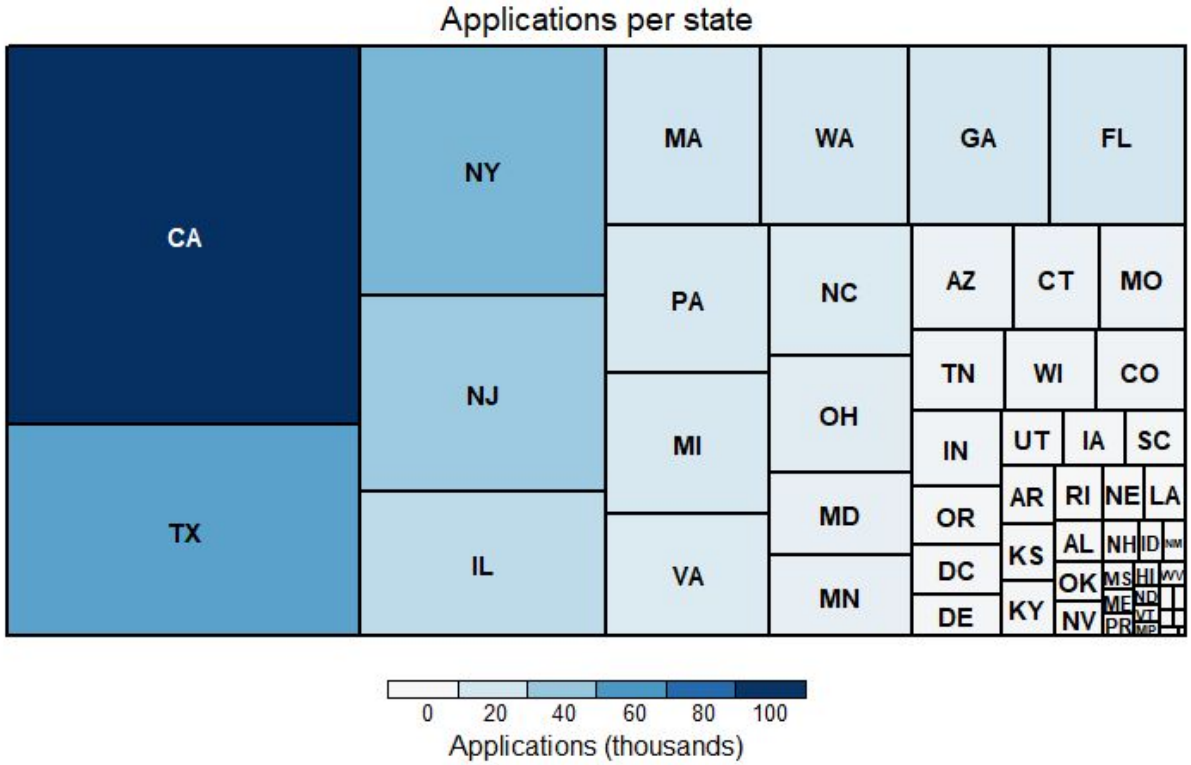


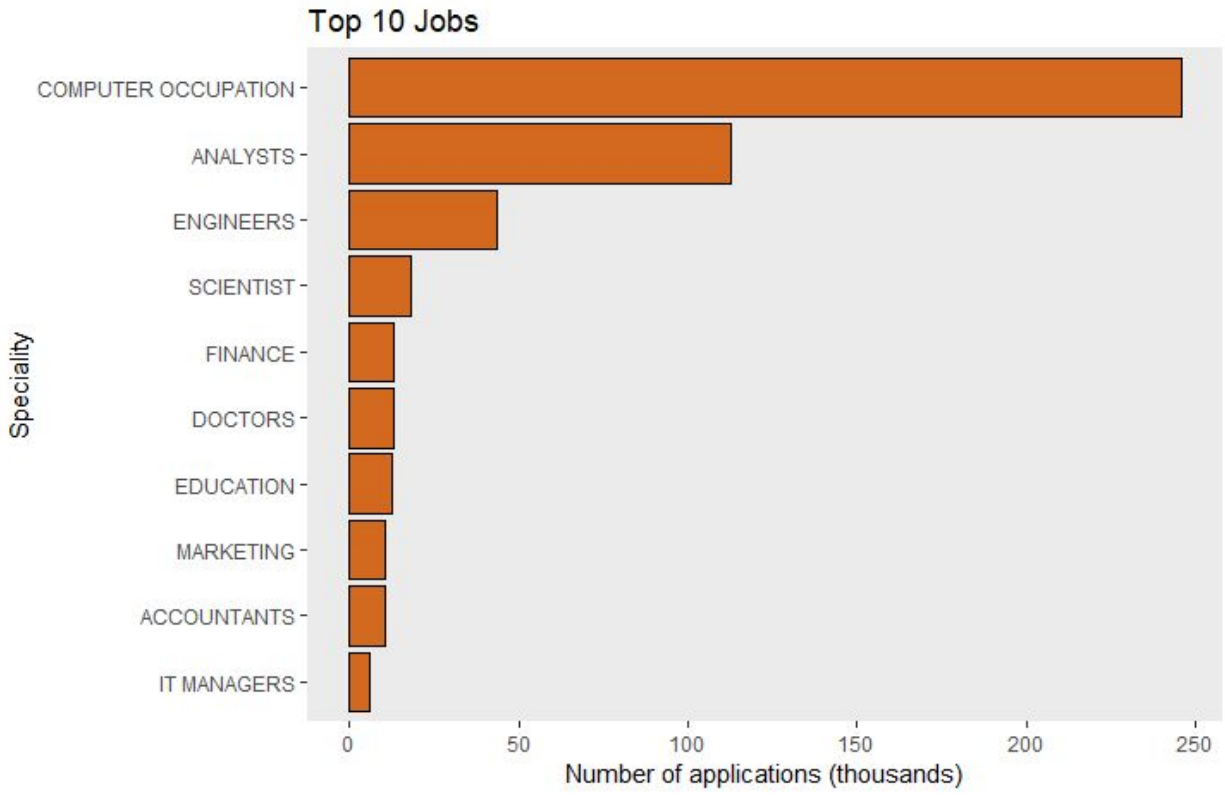

Balance the data:

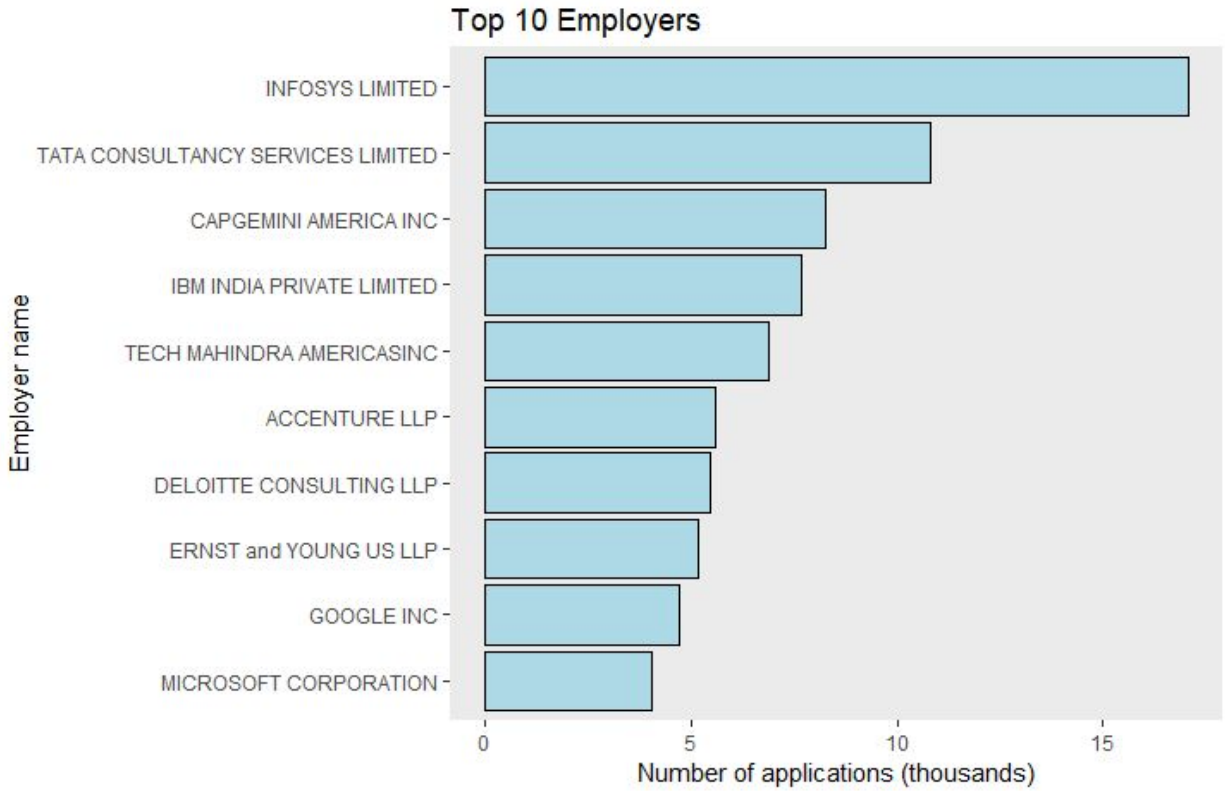
In our dataset, we have a very skewed response data, with “CERTIFIED” dominating (See graph “Visa Status”). Before we build and run our prediction models, we need first balance our response “CASE_STATUS”. After tried different tool, we decided to use SMOTE. This function handles unbalanced classification problems on our data. It can generate a new "SMOTE" (Supersampling Rare events) data set that addresses the class imbalance problem. Alternatively, it can also run a classification algorithm on this new data set and return the resulting model. (R documentation). The model gave us 142248 for certified and 140894 for denied. (Please see model building section for more information).

Visualizations of Raw Data:




Top 10 Jobs
COMPUTER OCCUPATION
ANALYSTS
ENGINEERS
SCIENTIST
FINANCE
DOCTORS
EDUCATION
MARKETING
ACCOUNTANTS
IT MANAGERS
Speciality
0
50
100
150
200
250
Number of applications (thousands)


Top 10 Employers
INFOSYS LIMITED
TATA CONSULTANCY SERVICES LIMITED
CAPGEMINI AMERICA INC
IBM INDIA PRIVATE LIMITED
TECH MAHINDRA AMERICASINC
ACCENTURE LLP
DELOITTE CONSULTING LLP
ERNST and YOUNG US LLP
GOOGLE INC
MICROSOFT CORPORATION
Employer name
0
5
10
15
Number of applications (thousands)

## Model Planning

Initial Hypotheses

1. The mean of wage offered is same for candidates whose visa has been approved is same as those candidates whose visa has been denied.

2. Employer_name has a significant relationship in case being approved or rejected.

3. The case status has no relationship with job title of the candidate.

Our goal is to predict if the mean of wage offered is same for candidates whose visa has been approved is same as those candidates whose visa has been denied. Predict if Employer_name has a significant relationship in case being approved or rejected. Predict if The case status has no relationship with job title of the candidate. We will also find whether there is any relationship between Case Status and other predictors. We will Code a model to Classify the Case status of application.

We will use Mean Squared Error (MSE) and Residual Mean Squared Error (RMSE) rates to compare performance of various models, with the following formulas:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y_i})^2 \qquad RMSE = \sqrt{MSE}$$

| | |
|---|---|
| Training data (80%) | 226515 |
| Test data (20%) | 56627 |

Table 1: Training and test data split

RMSE value is calculated by simply taking a square root of Mean squared error calculated before. We also use confusion matrix and ROC to compare our models.

We started building the models by first splitting the data into test and train sets. Our model had a total of 528134 observations or rows. The training set contains 80% of the data – 226515 observations, and test set contains 20% of the data – 56627 observations. Even though, our clean data set consisted of only 458,785 rows after removing the duplicates, we felt that train-test split is an effective method of validating our prediction model accurately.

After splitting the data into train and test sets, we decided to perform k = 5-fold Cross-Validation. We also created a 'ctrl' control parameter (see the R code attached) for using in Caret. We used Caret package in R, as it provides an easy uniform approach to fitting various models.

We did classification on our dataset. We had to approach different approaches for building models. Also manipulation of data for regression for fitting various models and getting the confusion matrix had to be done in independent ways

## Model Building

For classification we needed balanced data. After basic cleaning of data, we removed missing data and converted four levels of response into two levels as many models don't work with four levels. (We ran a PCA on our model to reduce the dimensionality of our data. Near zero variance and correlated predictors were removed as it would affect our models).

In order to remove unbalanced data we used SMOTE. This function handles unbalanced classification problems. It can generate a new "SMOTEd" (Supersampling Rare events) data set that addresses the class imbalance problem. Alternatively, it can also run a classification algorithm on this new data set and return the resulting model (R documentation). The model gave us 142248 for certified and 140894 for denied.

For balancing the data we also used ROSE (random over sampling examples). The package provides functions to deal with binary classification problems in the presence of imbalanced classes. Synthetic balanced samples are generated based on ROSE. Functions that implement more traditional remedies to the class imbalance are also provided, as well as different metrics to evaluate a learner accuracy. These are estimated by holdout, bootstrap or cross-validation methods. The model gave us 229106 for certified and 229679 for denied. However, we decided to proceed with SMOTE thought ROSE results were better because ROSE gave us a really different summary statistics with some variables which resulted in getting negative values after performing ROSE.

We also ran decision tree to improve our classification results. Implements Breiman's random forest algorithm (based on Breiman and Cutler's original Fortran code) for classification.

It can also be used in unsupervised mode for assessing proximities among data points. GLMNET was ran in order to know important variables in our model. We also got to know if there was any bias towards certain predictors. Various Classification models like logistic regression and Linear discriminant analysis.

Naïve Bayes is a model based on Bayes theorem that takes a probabilistic approach to selecting the attributes for the model.This approach produces a highly stable model with less variance. Naive Bayes algorithm assumes all attributes are independent of each other. It was built using case status as a dependent variable and all other predictors as independent variable. We received an accuracy of around 54% with the test data. Each of the models behaved differently with the data.

GLMNET was also built using case status as a dependent variable and all other predictors as independent variable. We received an accuracy of around 87% with the test data.

GLM was built using case status as a dependent variable and all other predictors as independent variable. We received an accuracy of around 87% with the test data. We felt that GLM and GLMNET behaved similarly.

Decision trees gave us an accuracy if 74.27% with the test data and it was built using case status as a dependent variable and all other predictors as independent variable similar to other models.

| **Models** | **Accuracy** |
|---|---|
| GLM | 86.47% |
| Boosted Logistic Regression | 83.12% |
| GLMNET | 87.06% |

| | |
|---|---|
| Decision Tree | 74.27% |
| Naive Bayes | 54.74% |
| Linear Discriminant model | 86.47% |

## Results and Performance

To evaluate the results and model performance, we compared it with baseline results.Then, we selected the best model and interpreted the parameters and drew conclusion from the results. In Balanced accuracy GLM outperformed other models. Naive Bayes seemed to perform the worst in terms of balanced accuracy. Please see below for Decision Tree and ROC curves.

| **Testing Performance** | **Specitivity** | **Sensitivity** | **Balanced Accuracy** |
|---|---|---|---|
| GLM | 0.8482 | 0.9020 | 0.8752 |
| LDA | 0.8280 | 0.9011 | 0.8647 |
| Decision Trees | 0.8327 | 0.7804 | 0.8064 |
| Naive Bayes | 0.0000 | 1.0000 | 0.5000 |
| GLMNET | 0.8429 | 0.8980 | 0.8704 |
| Log Boost | 0.8713 | 0.8020 | 0.8365 |

Generalized Linear Model

| Prediction | Certified | Denied |
|---|---|---|
| Certified | 25661 | 4277 |
| Denied | 2788 | 23901 |

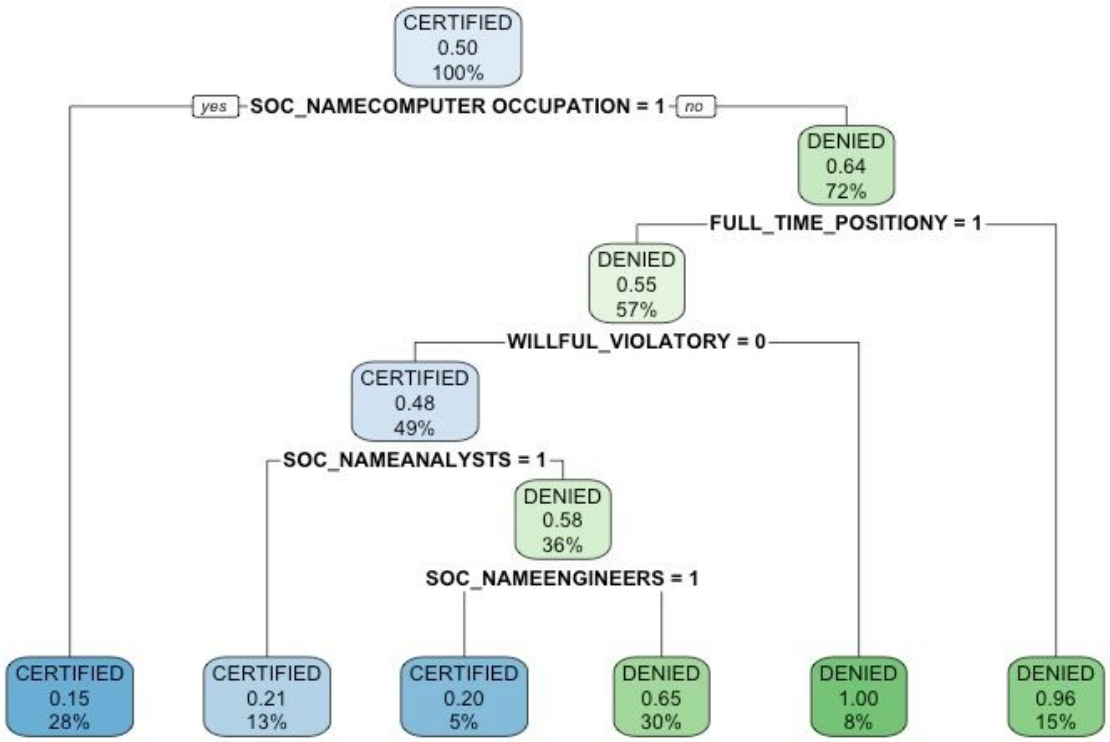
Accuracy : 0.8752

Linear Discriminant Model

| Prediction | Certified | Denied |
|---|---|---|
| Certified | 2635 | 4848 |
| Denied | 2814 | 23330 |

Accuracy : 0.8647

Decision Tree




| Prediction | Certified | Denied |
|---|---|---|
| Certified | 22203 | 4715 |
| Denied | 6246 | 23463 |

Accuracy : 0.8064

Logistic Boost

| Prediction | Certified | Denied |
|---|---|---|
| Certified | 22815 | 3627 |
| Denied | 5634 | 24551 |

Accuracy : 0.8365

GLMNET

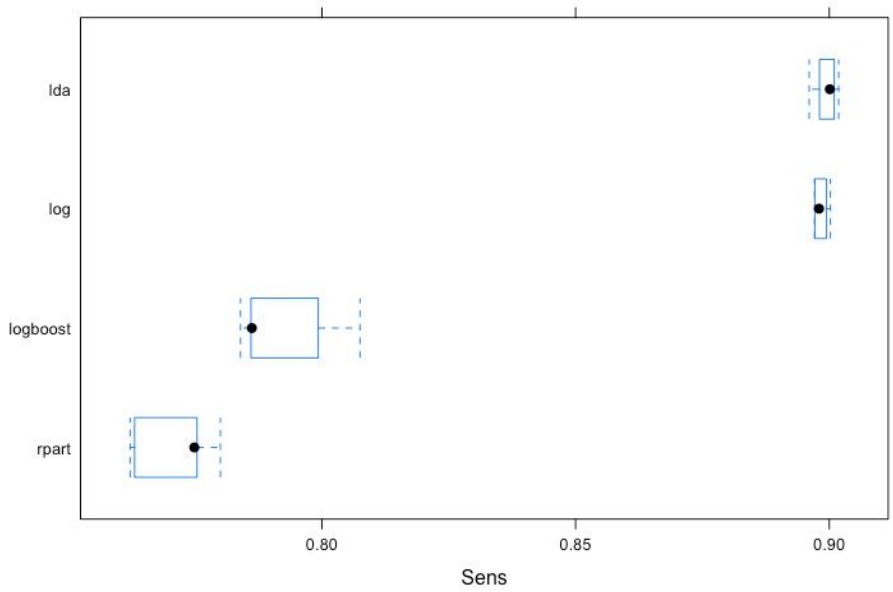
| Prediction | Certified | Denied |
|---|---|---|
| Certified | 25548 | 4428 |
| Denied | 2901 | 23750 |

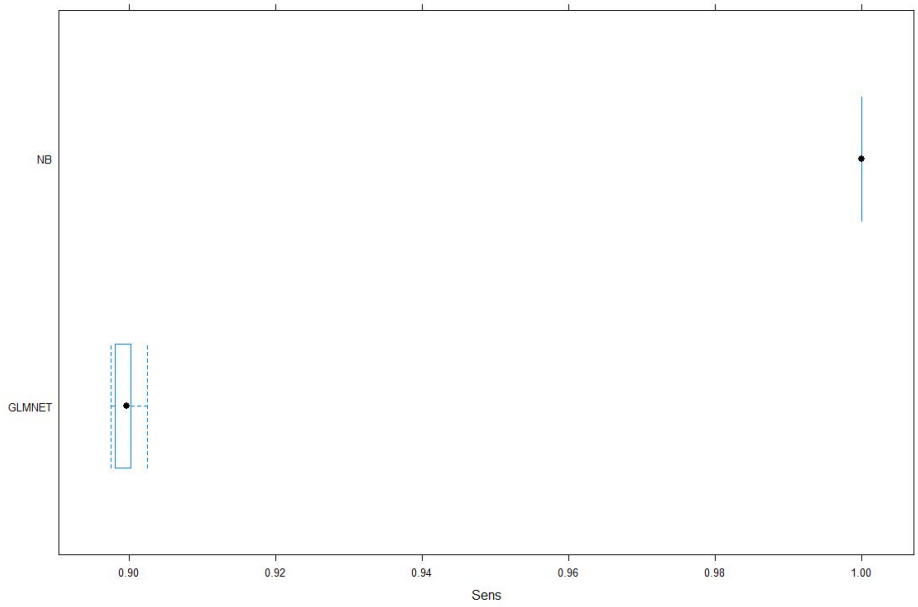
Accuracy : 0.8706

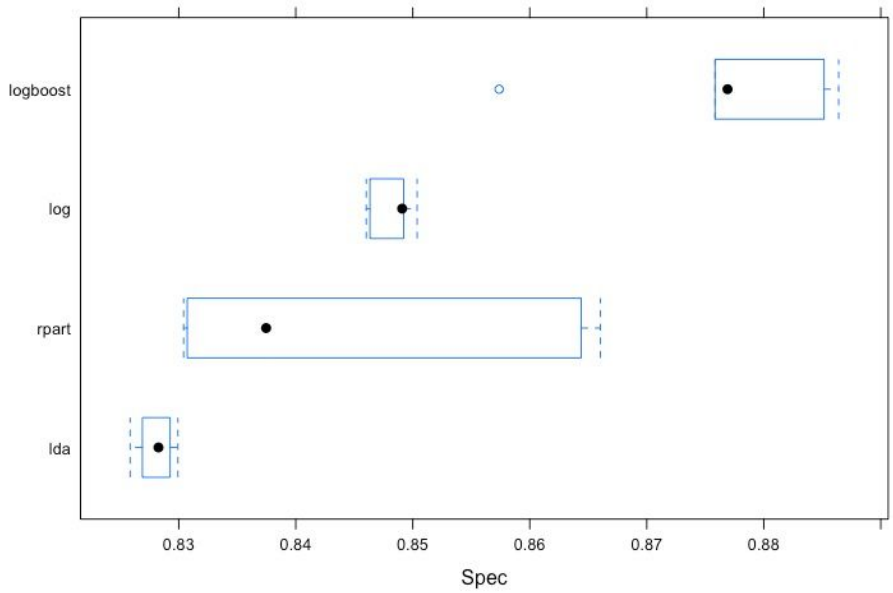
Naive Bayes

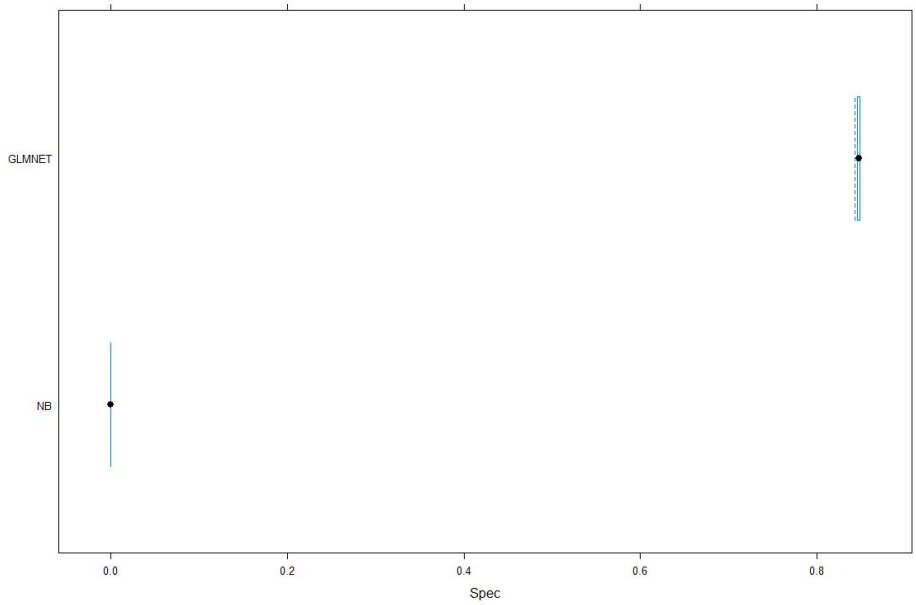
| Prediction | Certified | Denied |
|---|---|---|
| Certified | 40642 | 33597 |
| Denied | 0 | 0 |

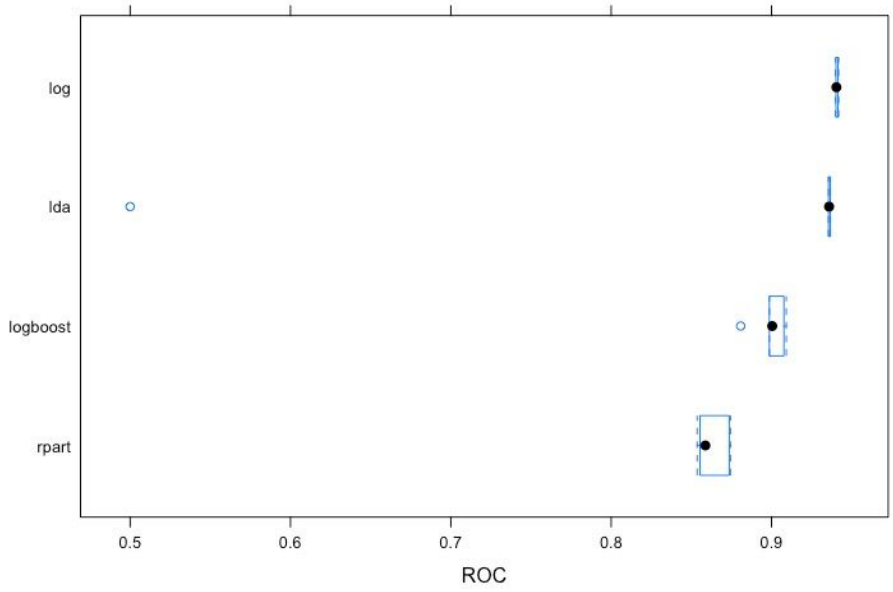
Accuracy : 0.5474

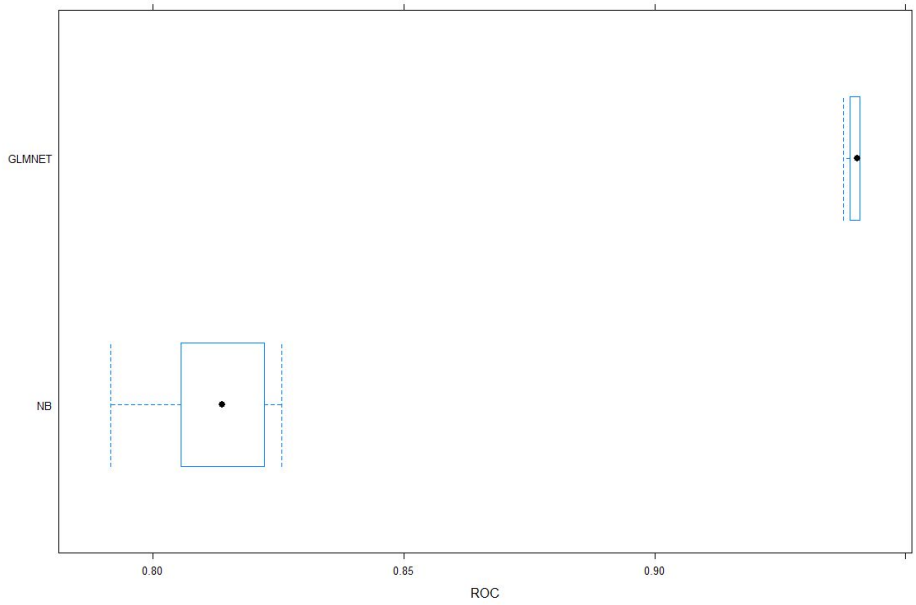

lda
log
logboost
rpart
0.80
0.85
0.90
Sens

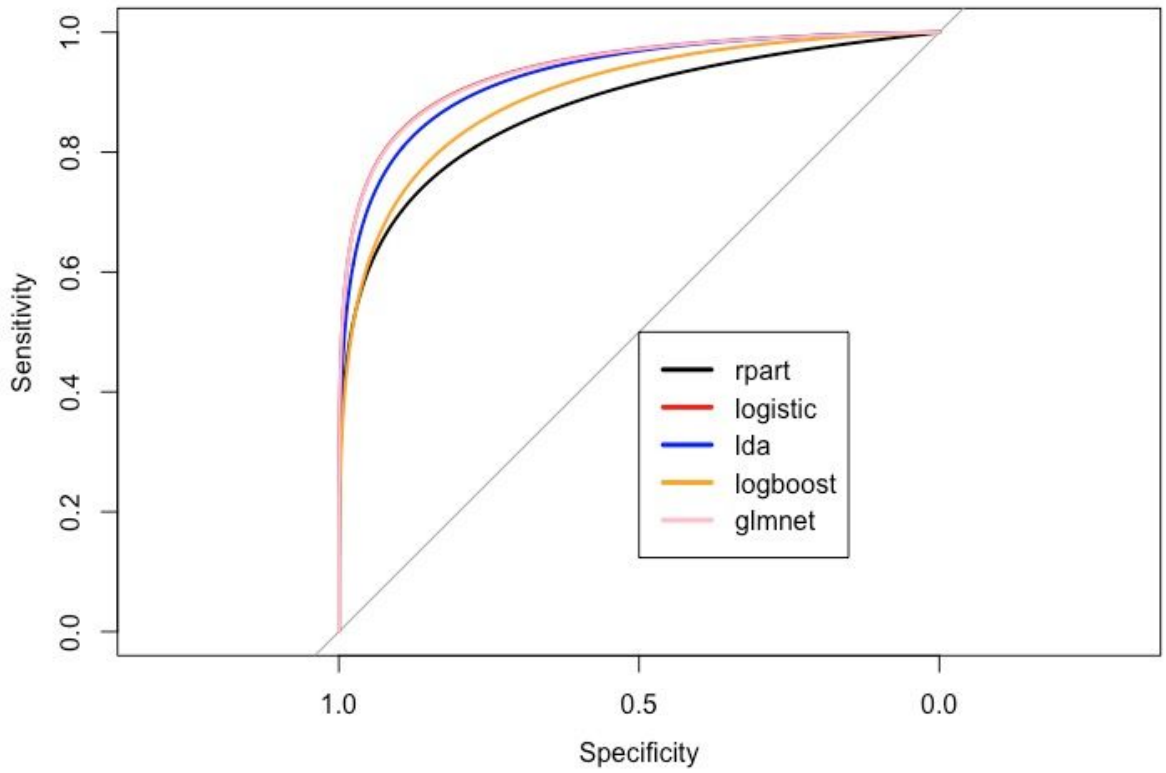

NB
GLMNET
0.90
0.92
0.94
0.96
0.98
1.00
Sens


logboost
log
rpart
lda
0.83
0.84
0.85
0.86
0.87
0.88
Spec


GLMNET
NB
0.0
0.2
0.4
0.6
0.8
Spec


log
lda
logboost
rpart
0.5
0.6
0.7
0.8
0.9
ROC


GLMNET
NB
0.80
0.85
0.90
ROC


Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.5
0.0
Specificity
rpart
logistic
lda
logboost
glmnet

## Discussions and Recommendations

Based on our accuracy and variance analysis, we observe that GLM is the best model for our dataset providing the highest accuracy and lowest variance. Naive Bayes has the lowest accuracy and highest variance among all three algorithms. According to our research, the important predictors SOC_NAME, WILLFUL_VIOLATOR, FULL_TIME_POSITIONS received more percentage than the others and are considered important factors in determining the case status of the Labor Condition Application filed by a perspective H1B employer to employ non-immigrant workers.

We feel undergraduate students or people looking to apply for H1B in the future could use our analysis to be sure to have their application certified. Some points noted were computer science, engineering and analyst occupation with full time position was the most popular ones in the case approved category. So we can say that job status is an important predictor in case being certified.

On April 18, 2017, President Trump signed the “Buy American and Hire American Executive Order” , which has brought uncertainty to the process of H1B. In the future, we would do more research to compare the disclosure data before and after the executive order, to see what differences it leads.

## References & Outputs

```
> varImp(h1b.log)
glm variable importance

  only 20 most important variables shown (out of 172)

                                Overall
FULL_TIME_POSITIONY                100.00
WILLFUL_VIOLATORY                   43.53
H.1B_DEPENDENTY                    40.48
SOC_NAMESCIENTIST                   38.95
PREVAILING_WAGE                    35.41
SOC_NAMEHEALTHCARE                  32.38
DECISION_MONTH                     30.38
CASE_SUBMITTED_MONTH                29.62
SOC_NAMECOUNSELORS                  29.57
SOC_NAMEMARKETING                   29.17
DECISION_YEAR                      28.75
CASE_SUBMITTED_YEAR                 27.45
`SOC_NAMEWRITERS EDITORS AND AUTHORS`       25.69
`SOC_NAMEMATHEMATICIANS AND STATISTICIANS`  25.46
`SOC_NAMEPUBLIC RELATIONS`          25.23
SOC_NAMEDOCTORS                    24.68
`SOC_NAMEMULTIMEDIA ARTISTS AND ANIMATORS`  19.67
`SOC_NAMESALES AND RELATED WORKERS`         19.60
`SOC_NAMECOMPUTER OCCUPATION`       15.94
`SOC_NAMEGRAPHIC DESIGNERS`         14.40
> getTrainPerf(h1b.log)
  TrainROC TrainSens TrainSpec method
1 0.9407383 0.8983822 0.8482114    glm
> h1b.log
Generalized Linear Model

226515 samples
   11 predictor
    2 classes: 'CERTIFIED', 'DENIED'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 181212, 181212, 181213, 181212, 181211
Resampling results:

  ROC       Sens      Spec
  0.9407383  0.8983822  0.8482114

> p.log <- predict(h1b.log, h1b.test)
```

```
> confusionMatrix(p.log, h1b.test$CASE_STATUS)
Confusion Matrix and Statistics

          Reference
Prediction  CERTIFIED DENIED
  CERTIFIED     25661   4277
  DENIED         2788  23901

               Accuracy : 0.8752
                 95% CI : (0.8725, 0.8779)
    No Information Rate : 0.5024
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7504
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9020
            Specificity : 0.8482
         Pos Pred Value : 0.8571
         Neg Pred Value : 0.8955
             Prevalence : 0.5024
         Detection Rate : 0.4532
   Detection Prevalence : 0.5287
      Balanced Accuracy : 0.8751

       'Positive' Class : CERTIFIED
```

LDA

```
> varImp(h1b.lda)
ROC curve variable importance

                    Importance
SOC_NAME               100.000
FULL_TIME_POSITION      52.511
WILLFUL_VIOLATOR        33.559
PREVAILING_WAGE         32.071
H.1B_DEPENDENT          19.106
DECISION_MONTH          10.949
CASE_SUBMITTED_MONTH     6.501
CASE_SUBMITTED_YEAR      3.943
DECISION_YEAR            1.881
WORKSITE_STATE           1.602
EMPLOYER_STATE           0.000
> getTrainPerf(h1b.lda)
  TrainROC TrainSens TrainSpec method
1 0.8489527 0.8995266 0.8280749    lda
> h1b.lda
Linear Discriminant Analysis

226515 samples
   11 predictor
    2 classes: 'CERTIFIED', 'DENIED'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 181212, 181212, 181213, 181212, 181211
Resampling results:

  ROC        Sens       Spec
  0.8489527  0.8995266  0.8280749

> p.lda <- predict(h1b.lda, h1b.test)
> confusionMatrix(p.lda, h1b.test$CASE_STATUS)
Confusion Matrix and Statistics

          Reference
Prediction  CERTIFIED DENIED
  CERTIFIED     25635   4848
  DENIED         2814  23330

               Accuracy : 0.8647
                 95% CI : (0.8618, 0.8675)
    No Information Rate : 0.5024
    P-Value [Acc > NIR] : < 2.2e-16
```

Kappa : 0.7293
Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9011
Specificity : 0.8280
Pos Pred Value : 0.8410
Neg Pred Value : 0.8924
Prevalence : 0.5024
Detection Rate : 0.4527
Detection Prevalence : 0.5383
Balanced Accuracy : 0.8645

'Positive' Class : CERTIFIED

Rpart:

Confusion Matrix and Statistics

```
          Reference
Prediction  CERTIFIED DENIED
  CERTIFIED     22203   4715
  DENIED         6246  23463

               Accuracy : 0.8064
                 95% CI : (0.8032, 0.8097)
    No Information Rate : 0.5024
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.613
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7804
            Specificity : 0.8327
         Pos Pred Value : 0.8248
         Neg Pred Value : 0.7898
             Prevalence : 0.5024
         Detection Rate : 0.3921
   Detection Prevalence : 0.4754
      Balanced Accuracy : 0.8066

       'Positive' Class : CERTIFIED
```

LogBoost

```
> varImp(h1b.boost)
ROC curve variable importance

                    Importance
SOC_NAME               100.000
FULL_TIME_POSITION      52.511
WILLFUL_VIOLATOR        33.559
PREVAILING_WAGE         32.071
H.1B_DEPENDENT          19.106
DECISION_MONTH          10.949
CASE_SUBMITTED_MONTH     6.501
CASE_SUBMITTED_YEAR      3.943
DECISION_YEAR            1.881
WORKSITE_STATE           1.602
EMPLOYER_STATE           0.000
> getTrainPerf(h1b.boost)
  TrainROC TrainSens TrainSpec     method
1 0.8993559 0.7925993 0.8763351 LogitBoost
> h1b.boost
Boosted Logistic Regression

226515 samples
   11 predictor
    2 classes: 'CERTIFIED', 'DENIED'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 181212, 181212, 181213, 181212, 181211
Resampling results across tuning parameters:

 nIter  ROC       Sens      Spec
 11     0.8686062  0.7882671  0.8639678
 21     0.8991918  0.7860263  0.8752173
 31     0.8993559  0.7925993  0.8763351

ROC was used to select the optimal model using the largest value.
The final value used for the model was nIter = 31.
> p.boost <- predict(h1b.boost, h1b.test)
> confusionMatrix(p.boost, h1b.test$CASE_STATUS)
Confusion Matrix and Statistics

          Reference
Prediction  CERTIFIED DENIED
 CERTIFIED     22815   3627
 DENIED         5634  24551
```

```
               Accuracy : 0.8365
                 95% CI : (0.8334, 0.8395)
    No Information Rate : 0.5024
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.673
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8020
            Specificity : 0.8713
         Pos Pred Value : 0.8628
         Neg Pred Value : 0.8134
             Prevalence : 0.5024
         Detection Rate : 0.4029
   Detection Prevalence : 0.4670
      Balanced Accuracy : 0.8366

       'Positive' Class : CERTIFIED
```

```
Naive Bayes
> h1B.nb
Naive Bayes

296962 samples
   11 predictor
    2 classes: 'CERTIFIED', 'DENIED'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 237569, 237569, 237570, 237570, 237570
Resampling results across tuning parameters:

  usekernel  ROC        Sens  Spec
  FALSE      0.0000000  NaN           NaN
   TRUE      0.5978864    1   7.440753e-06

Tuning parameter 'fL' was held constant at a value of 0
Tuning parameter 'adjust' was held constant
 at a value of 1
ROC was used to select the optimal model using the largest value.
The final values used for the model were fL = 0, usekernel = TRUE and adjust = 1.
> getTrainPerf(h1B.nb)
   TrainROC TrainSens    TrainSpec method
1 0.5978864         1 7.440753e-06     nb
> varImp(h1B.nb)
ROC curve variable importance

                   Importance
SOC_NAME              100.0000
FULL_TIME_POSITION     52.2689
WILLFUL_VIOLATOR       33.7945
PREVAILING_WAGE        32.8095
H.1B_DEPENDENT         18.6206
DECISION_MONTH         10.9418
CASE_SUBMITTED_MONTH    6.5197
CASE_SUBMITTED_YEAR     3.6417
DECISION_YEAR           1.0701
WORKSITE_STATE          0.7918
EMPLOYER_STATE          0.0000
> plot(h1B.nb)
> p.nb<- predict(h1B.nb,h1BTest)

Confusion Matrix and Statistics

            Reference
Prediction  CERTIFIED DENIED
  CERTIFIED      40642  33597
```

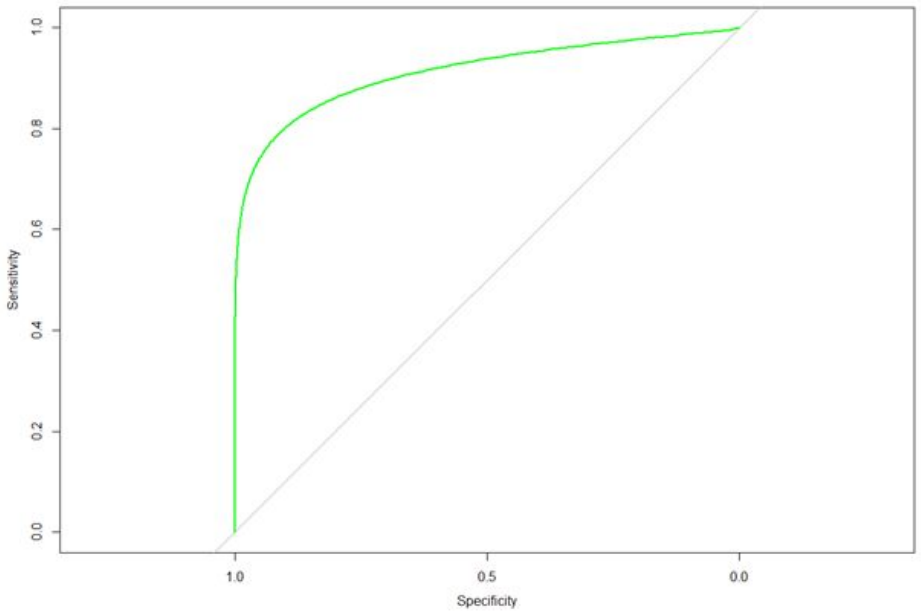
DENIED 0 0

Accuracy : 0.5474
95% CI : (0.5439, 0.551)
No Information Rate : 0.5474
P-Value [Acc > NIR] : 0.5015

Kappa : 0
Mcnemar's Test P-Value : <2e-16

Sensitivity : 1.0000
Specificity : 0.0000
Pos Pred Value : 0.5474
Neg Pred Value : NaN
Prevalence : 0.5474
Detection Rate : 0.5474
Detection Prevalence : 1.0000
Balanced Accuracy : 0.5000

'Positive' Class : CERTIFIED

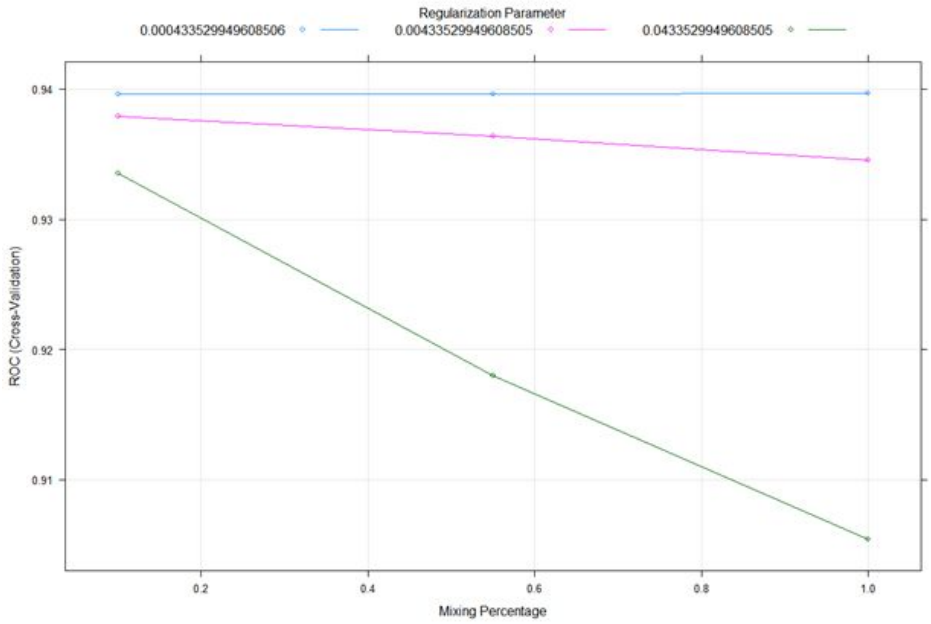


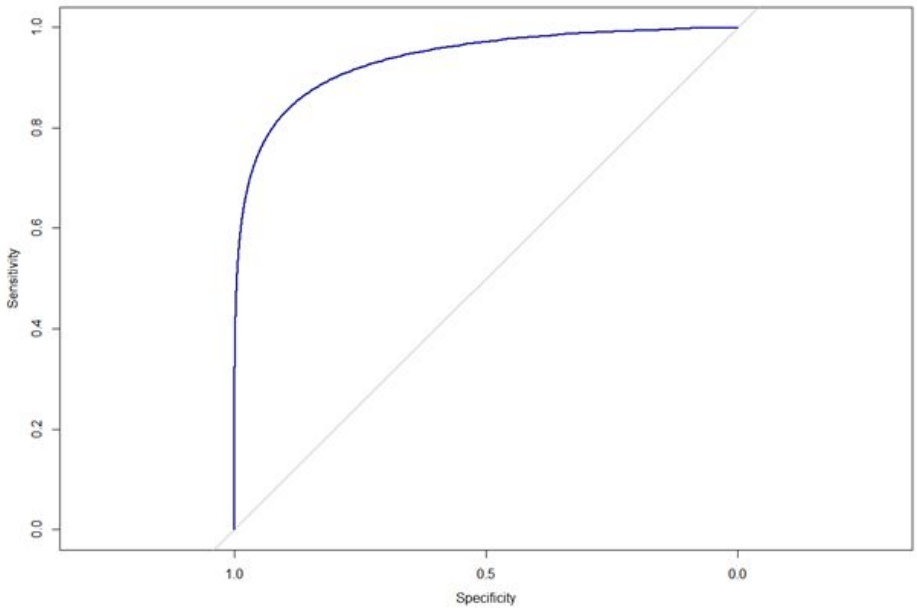

> auc(test.h1bnb.roc)
Area under the curve: 0.9504
> h1B.glmnet
glmnet

226515 samples

```
11 predictor
2 classes: 'CERTIFIED', 'DENIED'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 181212, 181212, 181212, 181211, 181213
Resampling results across tuning parameters:

 alpha  lambda       ROC        Sens       Spec
 0.10   0.0004335299  0.9395876  0.8993928  0.8468452
 0.10   0.0043352995  0.9379205  0.8987513  0.8440328
 0.10   0.0433529950  0.9335305  0.8969499  0.8354981
 0.55   0.0004335299  0.9396355  0.8994631  0.8466411
 0.55   0.0043352995  0.9363886  0.8987513  0.8410164
 0.55   0.0433529950  0.9179806  0.9015721  0.8085188
 1.00   0.0004335299  0.9396903  0.8996125  0.8464637
 1.00   0.0043352995  0.9345670  0.8992610  0.8386210
 1.00   0.0433529950  0.9054094  0.7603845  0.8757940

ROC was used to select the optimal model using the largest value.
The final values used for the model were alpha = 1 and lambda = 0.0004335299.
> varImp(h1B.glmnet)
glmnet variable importance

  only 20 most important variables shown (out of 172)

                                        Overall
WILLFUL_VIOLATORY                       100.00
SOC_NAMECOUNSELORS                      73.53
SOC_NAMEREAL ESTATE                     67.28
SOC_NAMEREPORTERS AND CORRESPONDENTS    65.92
SOC_NAMEPUBLIC RELATIONS                65.76
SOC_NAMEINSURANCE                       64.65
DECISION_YEAR                           63.71
SOC_NAMEINTERPRETERS AND TRANSLATORS    61.06
SOC_NAMESALES AND RELATED WORKERS       60.65
SOC_NAMEHEALTHCARE                      57.68
CASE_SUBMITTED_YEAR                     57.10
SOC_NAMEMULTIMEDIA ARTISTS AND ANIMATORS  51.32
SOC_NAMEWRITERS EDITORS AND AUTHORS     50.79
FULL_TIME_POSITIONY                     50.67
SOC_NAMEEVENT PLANNERS                  50.51
SOC_NAMEENGINEERS                       50.18
SOC_NAMEFOOD PREPARATION WORKERS        49.53
SOC_NAMESCIENTIST                       48.10
SOC_NAMEMECHANICS                       47.80
SOC_NAMEFIRST LINE SUPERVISORS          42.11
> GLMNET <- save(h1B.glmnet, file = "h1B.glmnet.rda")
```

```
> getTrainPerf(h1B.glmnet)
   TrainROC TrainSens TrainSpec method
1 0.9396903 0.8996125 0.8464637 glmnet

Confusion Matrix and Statistics

          Reference
Prediction  CERTIFIED DENIED
  CERTIFIED      25548   4428
  DENIED          2901  23750

          Accuracy : 0.8706
          95% CI : (0.8678, 0.8733)
          No Information Rate : 0.5024
          P-Value [Acc > NIR] : < 2.2e-16

          Kappa : 0.7411
 Mcnemar's Test P-Value : < 2.2e-16

          Sensitivity : 0.8980
          Specificity : 0.8429
          Pos Pred Value : 0.8523
          Neg Pred Value : 0.8911
          Prevalence : 0.5024
          Detection Rate : 0.4512
    Detection Prevalence : 0.5294
          Balanced Accuracy : 0.8704

          'Positive' Class : CERTIFIED
```




```
> summary(rValues)

Call:
summary.resamples(object = rValues)

Models: rpart, log, logboost, lda
Number of resamples: 5

ROC
          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
```

```
rpart    0.8536175 0.8553934 0.8587340 0.8631549 0.8736353 0.8743940    0
log      0.9397410 0.9401934 0.9404464 0.9407383 0.9414539 0.9418568    0
logboost 0.8807888 0.8986433 0.9004336 0.8993559 0.9077197 0.9091942    0
lda      0.5000000 0.9355233 0.9359054 0.8489527 0.9366539 0.9366811    0

Sens
             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
rpart    0.7622144 0.7630931 0.7748682 0.7711141 0.7753954 0.7799991    0
log      0.8971002 0.8971441 0.8979789 0.8983822 0.8994683 0.9002197    0
logboost 0.7839631 0.7860281 0.7862039 0.7925993 0.7992882 0.8075132    0
lda      0.8960457 0.8990773 0.9001076 0.8995266 0.9005569 0.9018453    1

Spec
             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
rpart    0.8304130 0.8307235 0.8374662 0.8458070 0.8643985 0.8660338    0
log      0.8460276 0.8463381 0.8490951 0.8482114 0.8492215 0.8503748    0
logboost 0.8573837 0.8758373 0.8769019 0.8763351 0.8851528 0.8863999    0
lda      0.8258439 0.8274133 0.8282654 0.8280749 0.8289269 0.8299250    1
```

**R codes:**

```
library("caret")
library("randomForest")
library("dplyr")
library("glmnet")
library("ROCR")
library("DMwR")
library("ROSE")
library("pROC")
library("rpart.plot")

h1b <- read.csv('1. Master H1B Dataset.csv')
h1b <- na.omit(h1b)
str(h1b)

h1b <- subset(h1b, select = -c(CASE_SUBMITTED_DAY, DECISION_DAY, EMPLOYER_NAME,
NAICS_CODE, PW_SOURCE,
                      PW_SOURCE_YEAR, PW_SOURCE_OTHER, WAGE_RATE_OF_PAY_FROM,
WAGE_RATE_OF_PAY_TO,
                      WAGE_UNIT_OF_PAY, WORKSITE_POSTAL_CODE))

h1b <- subset(h1b, h1b$VISA_CLASS=="H1B")
h1b <- subset(h1b, h1b$EMPLOYER_COUNTRY=="UNITED STATES OF AMERICA")
h1b <- subset(h1b, select = -c(VISA_CLASS, EMPLOYER_COUNTRY))
h1b <- subset(h1b, h1b$TOTAL_WORKERS=='1')
h1b <- subset(h1b, select = -c(TOTAL_WORKERS))
h1b <- subset(h1b, h1b$CASE_STATUS != "WITHDRAWN")
h1b$CASE_STATUS[h1b$CASE_STATUS == "CERTIFIEDWITHDRAWN"] <- "CERTIFIED"

h1b <- h1b[!(h1b$FULL_TIME_POSITION==""),]
h1b <- h1b[!(h1b$PW_UNIT_OF_PAY==""),]
h1b <- h1b[!(h1b$H.1B_DEPENDENT==""),]
h1b <- h1b[!(h1b$WILLFUL_VIOLATOR==""),]
h1b <- droplevels(h1b)

pw_unit_to_yearly <- function(prevailing_wage, pw_unit_of_pay) {
  return(ifelse(pw_unit_of_pay == "Year",
              prevailing_wage,
              ifelse(pw_unit_of_pay == "Hour",
                    2080*prevailing_wage,
                    ifelse(pw_unit_of_pay== "Week",
                          52*prevailing_wage,
                          ifelse(pw_unit_of_pay == "Month",
                                12*prevailing_wage,
                                26*prevailing_wage)))))
}
```

```
h1b %>%
  filter(!is.na(PW_UNIT_OF_PAY)) %>%
  mutate(PREVAILING_WAGE = as.numeric(PREVAILING_WAGE)) %>%
  mutate(PREVAILING_WAGE = pw_unit_to_yearly(PREVAILING_WAGE, PW_UNIT_OF_PAY)) %>%
  select(everything()) -> h1bconvert

summary(h1bconvert$PREVAILING_WAGE)
boxplot(h1bconvert$PREVAILING_WAGE)

sh1b <- subset(h1bconvert,h1bconvert$PREVAILING_WAGE<10000000)
boxplot(sh1b$PREVAILING_WAGE)

ssh1b <- subset(h1bconvert,h1bconvert$PREVAILING_WAGE<1000000)
boxplot(ssh1b$PREVAILING_WAGE)

h1bconvert <- h1bconvert[!(h1bconvert$PREVAILING_WAGE<10000 |
h1bconvert$PREVAILING_WAGE>400000),]
summary(h1bconvert$PREVAILING_WAGE)

h1bconvert <- subset(h1bconvert, select = -c(PW_UNIT_OF_PAY))
h1bclean <- h1bconvert

##dummy <- dummyVars( ~., data=h1bclean)
##dummy.h1b <- predict(dummy, newdata = h1bclean)

smote <- SMOTE(CASE_STATUS ~ ., data = h1bclean, perc.over = 2500, perc.under = 105)
table(smote$CASE_STATUS)
rose <- ROSE(CASE_STATUS ~ ., data = h1bclean)$data
table(rose$CASE_STATUS)

h1bsmote <- smote

ctrl <- trainControl(method="cv", number=5,
                     classProbs=TRUE,
                     #function used to measure performance
                     summaryFunction = twoClassSummary, #multiClassSummary for non binary
                     allowParallel =  TRUE)

trainIndex <- createDataPartition(h1bsmote$CASE_STATUS, p=.8, list=F)
h1b.train <- h1bsmote[trainIndex,]
h1b.test <- h1bsmote[-trainIndex,]

set.seed(199)
h1b.log <- train(CASE_STATUS ~ ., data=h1b.train, method="glm", family="binomial", metric="ROC",
trControl=ctrl)
varImp(h1b.log)
getTrainPerf(h1b.log)
```

```
h1b.log
p.log <- predict(h1b.log, h1b.test)
confusionMatrix(p.log, h1b.test$CASE_STATUS)

set.seed(199)
h1b.lda <-  train(CASE_STATUS ~ ., data=h1b.train, method="lda", family="binomial", metric="ROC",
trControl=ctrl)
varImp(h1b.lda)
getTrainPerf(h1b.lda)
h1b.lda
p.lda <- predict(h1b.lda, h1b.test)
confusionMatrix(p.lda, h1b.test$CASE_STATUS)

set.seed(199)
h1b.rpart <- train(CASE_STATUS ~ ., data=h1b.train, trControl = ctrl, metric="ROC", method="rpart")
p.rpart <- predict(h1b.rpart, h1b.test)
confusionMatrix(p.rpart, h1b.test$CASE_STATUS)
rpart.plot(h1b.rpart$finalModel)

set.seed(199)
h1b.boost <-  train(CASE_STATUS ~ ., data=h1b.train, method="LogitBoost", family="binomial", metric="ROC",
trControl=ctrl)
varImp(h1b.boost)
getTrainPerf(h1b.boost)
h1b.boost
p.boost <- predict(h1b.boost, h1b.test)
confusionMatrix(p.boost, h1b.test$CASE_STATUS)

rValues <- resamples(list(rpart=h1b.rpart, log=h1b.log, logboost=h1b.boost, lda=h1b.lda))
bwplot(rValues, metric="ROC")
bwplot(rValues, metric="Sens") #Sensitivity
bwplot(rValues, metric="Spec")
summary(rValues)


#NAIVE BAYES:
##Naive Bayes
modelLookup("nb") #we have some parameters to tune such as laplace correction
set.seed(192)
library(MLmetrics)
#h1B.nb <- train(CASE_STATUS ~ .-EMPLOYER_NAME - WORKSITE_POSTAL_CODE, data = h1BTrain,
#trControl = ctrl,
#metric = "ROC", #using AUC to find best performing parameters
#method = "nb")
h1B.nb <- train(CASE_STATUS ~ ., data = h1BTrain,
                trControl = ctrl,
                metric = "ROC", #using AUC to find best performing parameters
                method = "nb")
```

```
h1B.nb
getTrainPerf(h1B.nb)
varImp(h1B.nb)
plot(h1B.nb)
p.nb<- predict(h1B.nb,h1BTest)
confusionMatrix(p.nb, h1BTest$CASE_STATUS)
nb <- save(p.nb, file = "naivebayesh1b.rda")
load("naivebayesh1b.rda")

p.rpart2 <- predict(h1b.rpart, h1b.test, type="prob")
p.log2 <- predict(h1b.log, h1b.test, type="prob")
p.lda2 <- predict(h1b.lda, h1b.test, type="prob")
p.boost2 <- predict(h1b.boost, h1b.test, type="prob")

rpart.roc <- roc(h1b.test$CASE_STATUS, p.rpart2$DENIED)
log.roc <- roc(h1b.test$CASE_STATUS, p.log2$DENIED)
lda.roc <- roc(h1b.test$CASE_STATUS, p.lda2$DENIED)
glmnet.roc <- roc(h1b.test$CASE_STATUS, p.glmnet$DENIED)
#nb.roc <- roc(h1b.test$CASE_STATUS, p.nb$DENIED)
logboost.roc <- roc(h1b.test$CASE_STATUS, p.boost2$DENIED)

plot(smooth(rpart.roc), col="black")
plot(smooth(log.roc), add=T, col="red")
plot(smooth(lda.roc), add=T, col="blue")
plot(smooth(logboost.roc), add=T, col="orange")
plot(smooth(glmnet.roc), add=T, col="pink")
legend(x=.5, y=.5, cex=1, legend=c("rpart","logistic","lda","logboost","glmnet"), col=c("black", "red", "blue",
"orange", "pink"), lwd=3)

PCA
pca
library(AppliedPredictiveModeling)
#CASE_STATUS <- factor(c("CERTIFIED","WITHDRAWN"))

pp_hpc1 <- preProcess(h1b[, ],
                      method = c("center", "scale", "BoxCox"))
pp_hpc1 #can see results of processing

#now we just need to apply the processing model to the data
#transformed <- predict(pp_hpc, newdata = h1B[, -8,-26])
#head(transformed)

transformed1 <- predict(pp_hpc1, newdata = h1b[, ])
head(transformed1)

#numpending very low variance
mean(h1bdrop$NumPending == 0) #76%=0
```

```
#feature reduction using PCA

pp_no_pca1 <- preProcess(h1b[, ], method = c("pca"))
pp_no_pca1

head(predict(pp_no_pca1, newdata = h1b[ , ]))
finaldata <- (predict(pp_no_pca1, newdata = h1b[ , ]))
```