PREDICTING AND CLASSIFYING H1-B VISA APPLICATION STATUS

CAL STATE EAST BAY

CONTENTS

1. INTRODUCTION

The US H1-B visa is an employment-based non-immigrant visa that allows US companies to employ graduate level workers in specialty occupations that require theoretical or technical expertise in specialized fields such as IT, finance, accounting, architecture, engineering, mathematics, science, medicine, etc. Foreign workers fill a critical need in the U.S. labor market—particularly in the Science, Technology, Engineering, and Math (STEM) fields. Every year, U.S. employers seeking highly skilled foreign professionals submit their petitions for the pool of H-1B visa numbers for which U.S. Citizenship and Immigration Services (USCIS) controls the allocation. With a low statutory limit of visa numbers available, demand for H-1B visa numbers has outstripped the supply in recent years, and the cap has been reached quickly. Research shows that H-1B workers complement U.S. workers, fill employment gaps in many STEM occupations, and expand job opportunities for all.

This report demonstrates methods that can be used to statistically predict the outcome of H-1B applications before they are submitted to the USCIS. It highlights the relationships between different attributes, and examines them visually through use of charts. In addition, experiments are performed with a variety of data mining models and techniques and finally this report recommends a model to predict an application's outcome as approved or denied.

2. MOTIVATION AND PROBLEM STATEMENT

More than 50% of graduate students in STEM university programs are foreign students. They are the direct beneficiaries of the H1-b visa category which is specially designed for non-immigrant workers. As foreign students ourselves, we can relate to the challenges of getting an approved H1-b visa. For this project, we put ourselves in the role of a data analyst for a visa/immigration consultancy firm and attempted to create a machine learning model which could accurately predict the outcome of an application based on its attributes. This analysis will help foreign students and job seekers gain better insight of different factors that affect the Visa Application approval or denial. The main focus of the project is to find an algorithm that can accurately classify the denied visa applications, and subsequently prevent the loss of money to application fees.

Using various tools, we will be able to analyze and explain the following questions:

- Explore how different variables interact with CASE_STATUS approval

- Distribution of applications across regions based on prevailing wages
- Recommend clients whether the application will be APPROVE/DENIED

3. DESCRIPTION OF DATASET

Data Set: h1b_kaggle.csv
Source: https://www.kaggle.com/nsharan/h-1b-visa

The data used for the project has been picked from the Kaggle website. The raw data has been imported from The Office of Foreign Labor Certification (OFLC) website and a set of data transformations were performed to make the data more accessible for quick exploration.

This dataset contains five years' worth of H-1B petition data. The dataset has **3,002,458** rows and **11** variables over years 2011 to 2016. It includes the following attributes.

- **X:** Unique identifier assigned to each application submitted for processing.
- **CASE_STATUS:** Status associated with the last significant event or decision. Valid values include "Certified," "Certified-Withdrawn," Denied," and "Withdrawn".
- **EMPLOYER_NAME:** Name of employer submitting labor condition application.
- **SOC_NAME:** Occupational name associated with the requested job under temporary labor condition. Standard Occupational Classification system defines the codes and the names associated with them.
- **JOB_TITLE:** The requested job title in the petition.
- **FULL_TIME_POSITION:** Y = Full Time Position; N = Part Time Position
- **PREVAILING_WAGE:** Prevailing Wage for the job being requested for temporary labor condition.
- **YEAR:** Year when petition is filed (in between 2011-2016).
- **WORKSITE:** The address of the employer worksite.
- **Lon:** Longitude of the employer worksite.
- **Lat:** Latitude of the employer worksite.

4. DATA PRE-PROCESSING

The raw data contains 6 years' worth of data (2011-2016) which amounts to over 3 million records. Data is filtered out for the year 2016 for ease of analysis. Before preprocessing and data cleaning, our truncated dataset contains **647,803** records.

The percentage of missing values (N/A) is approx **2.7%** of records. These values have been excluded from the analysis as they can't be handled with any random values.

The outcome variable to be predicted is CASE_STATUS. The project tries to establish the dependency of the decision of CASE_STATUS on the attributes of the application. The existing distribution of different status labels in 2016 data is as follows.

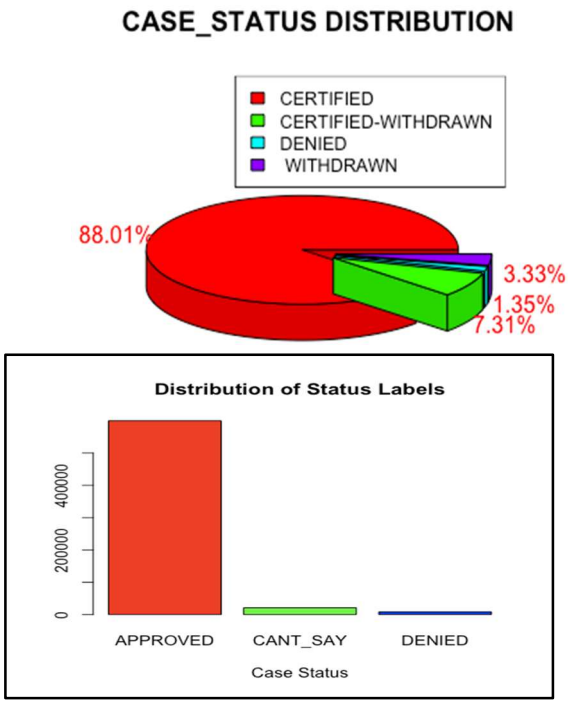


Fig 4.1: Distribution of case status

The aim is to predict the case decision as either "APPROVED" or "DENIED". Records are re-labeled with status as "WITHDRAWN" to "CANT_SAY", because these applicants have withdrawn their petitions before receiving any result from the labor department, hence predicting a decision in these cases won't be feasible. Further, data is merged for the statuses "CERTIFIED-WITHDRAWN" with "CERTIFIED" and defined as new case status "APPROVED".

The attribute “WORKSITE” contains the address of the employer worksite. It is being segregated into ‘State’ and ‘City’.

The data has been divided into 5 demographics regions based on longitude and latitude viz. **Midwest, Northeast, South, West and Others.**

In occupational roles (SOC_NAME), the data has large no. of similar values.The reference was taken from Govt. approved job categories and categorized all SOC names into 9 major job categories as: **Business, Computer, Education, Engineering, Healthcare, Management, Science, Arts and Others.**

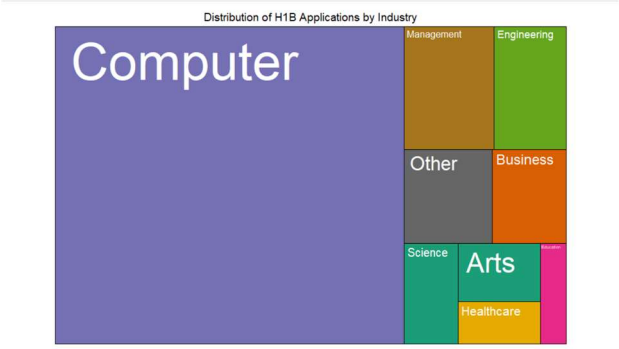
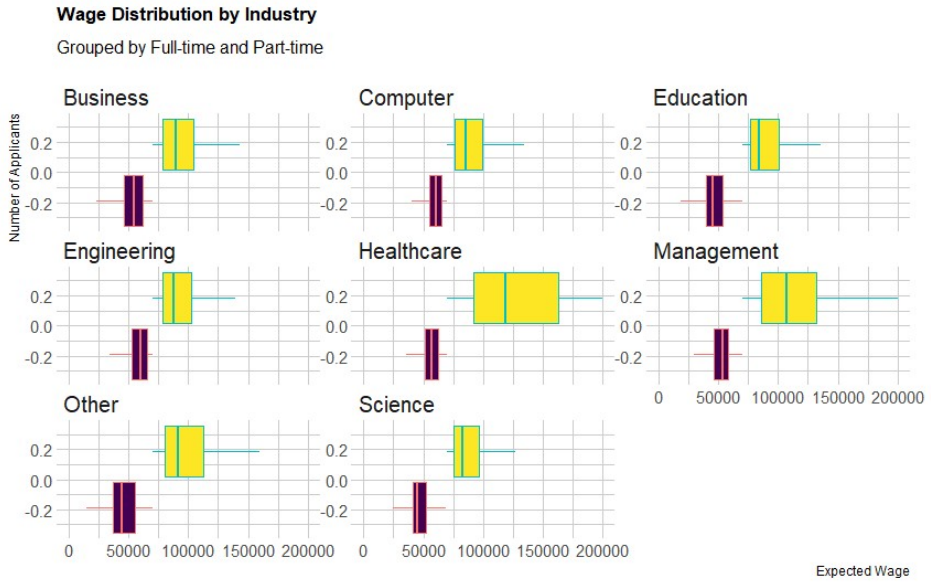


Fig 4.2: Treemap for category

5. EXPLORATORY ANALYSIS

Once preprocessing was performed, the rate of **Denied was discovered to be 1.3% of the dataset**. In addition, this report will explore the relationship between different variables, and how those interactions impact the approval outcome of the application. However, because the data is almost entirely categorical, a traditional correlation plot was not very useful for visual exploration.

The below chart shows boxplots which demonstrate the difference in salary between part time and full time employees. The Y axis simply describes the offset of the boxplots and can be ignored. It can be concluded that full time work has a wide range of values, while part-time work has a narrower range. In addition it can be seen that industries have significantly different outcomes on average wages.



The below chart presents the same information from above in an alternative format. From this view, the Y axis shows the count of workers and the X axis shows their wages. This graph displays the relationship and volume of full time to part time workers.

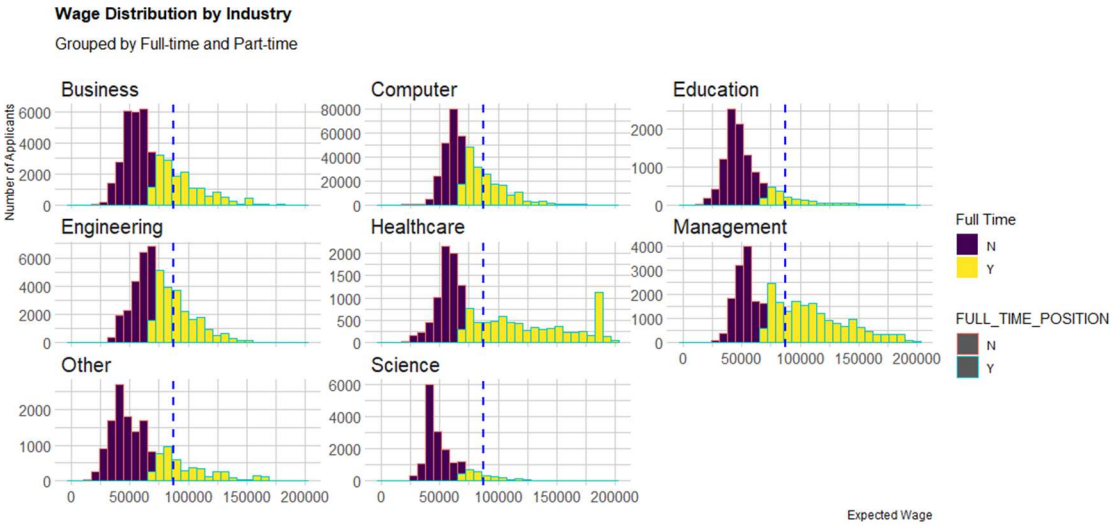
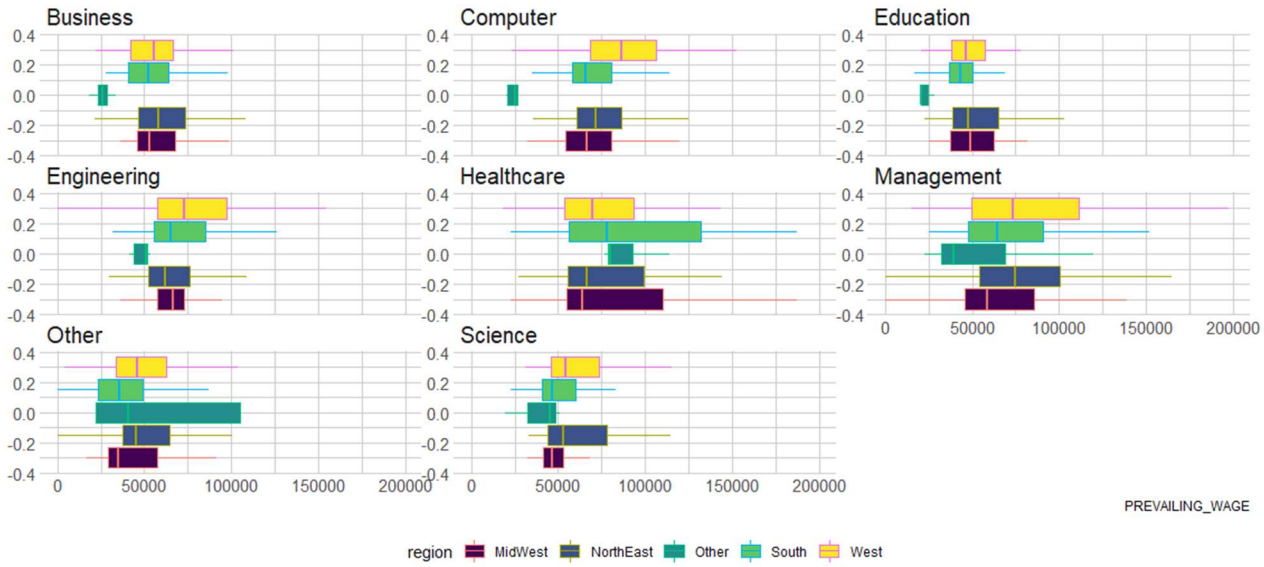
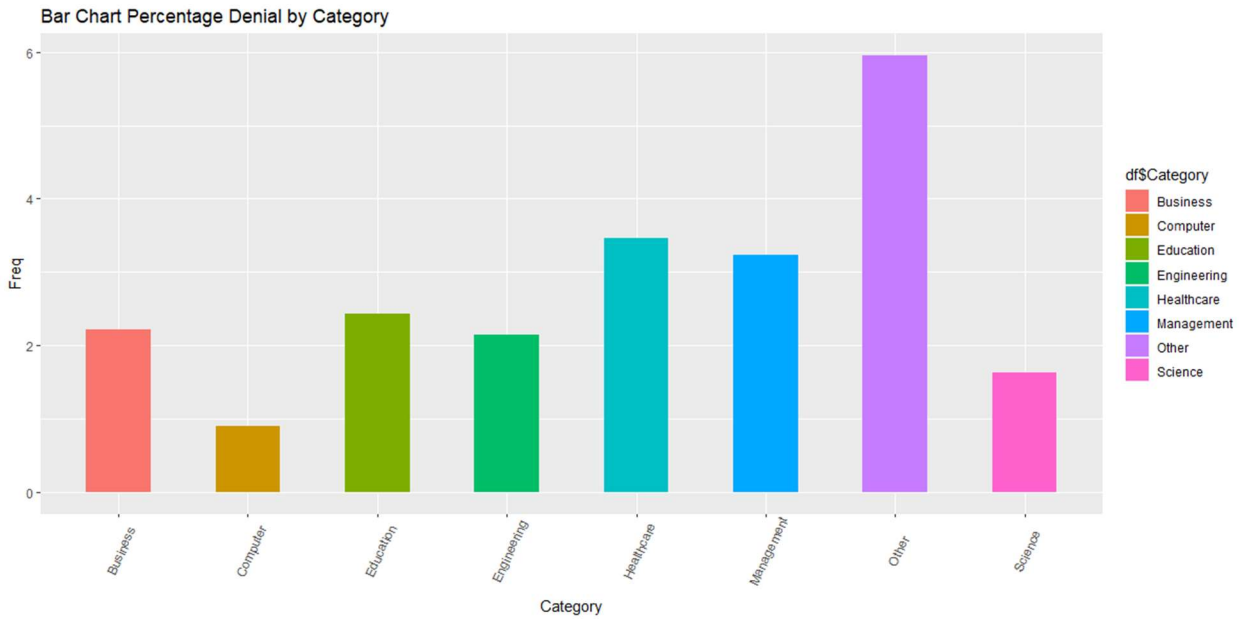


Fig 5.2 Wage distribution by

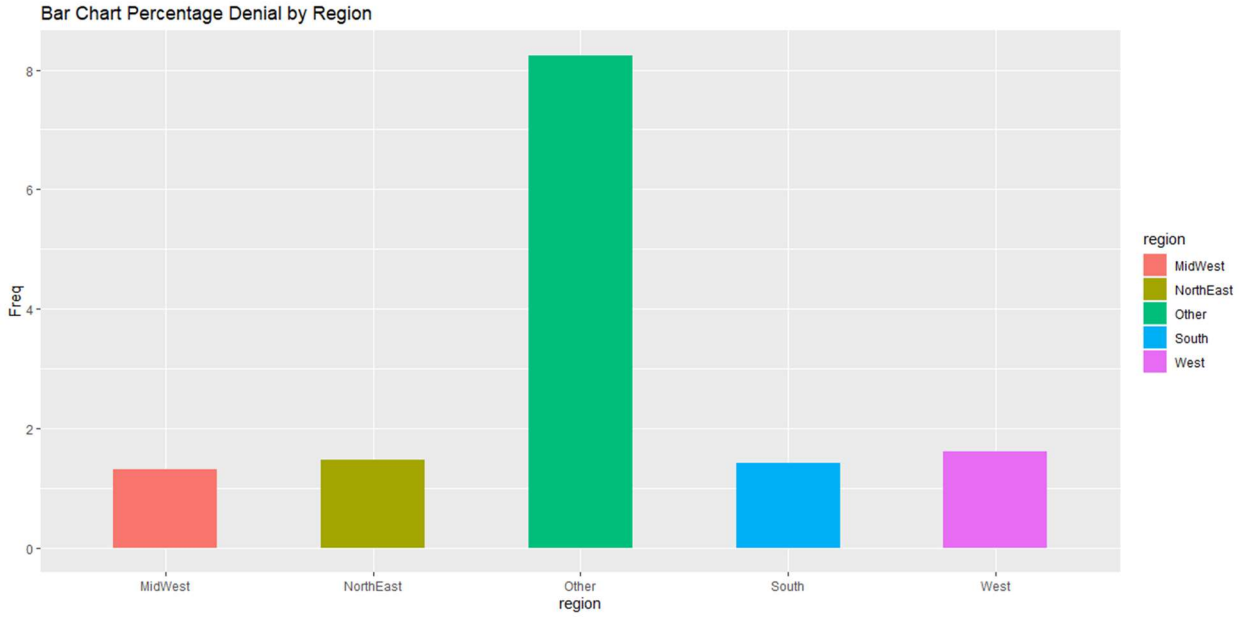
In the following chart the overall salary is compared by region and category. While the "Other" category for location varies wildly by salary, it is worth noting that the number of records from locations outside the continental United States was very small, and is prone to suffer from insufficient data. From this chart, it is clear that the region has a moderate influence on wage for certain categories of work. For example, 75% of computer jobs in the west pay more than the median wage for computer work in other regions. Another finding is that healthcare workers in the South have a wider range of high paying jobs than healthcare workers from other regions. A final result is that Science workers make decidedly less money working in the midwest than their counterparts in other regions.



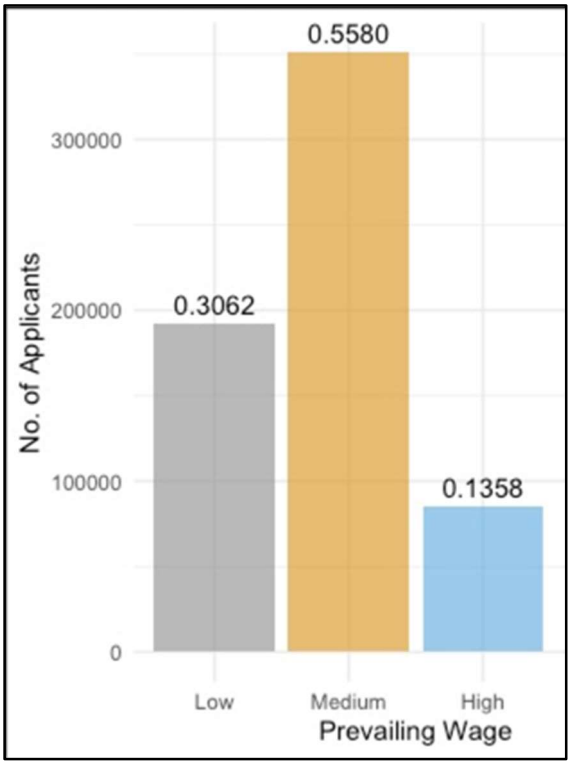
This report seeks to analyze the factors that have the greatest impact on application denial. The lower chart shows bar charts by the percentage of denial. The data shows that Computer is the working category with a significantly lower percentage of rejections at just under 1%. Meanwhile, “Other” work category has a far higher percentage of rejection than all other categories.



After examining the relationship between region and denial, it is clear that workers outside of the mainland United States suffer a far greater risk of denied applications.



By analysing the distribution of applicants over different ranges of prevailing wage, we can see that the majority of applicants are within the medium prevailing wage range of 60k-100k.



6. DATA MINING METHODS

After completing all the exploratory analysis and preprocessing of the data, The predictor selected for producing models using different algorithms are: **FULL_TIME_POSITION, PREVAILING_WAGE, lon, lat, STATUS, Category.**

- **KNN, Classification Tree, Neural Net** which are Machine learning data-driven techniques and are non parametric method, as no assumption is made and no parameters are evaluated
- **Logistic regression**: Model driven and assumption based supervised learning technique used only for classifying categorical variables.
- The performance (accuracy) of all the models are checked by using the validation data.

6.1 MODELS WITH ORIGINAL DATASET

6.1.1. KNN

Dummies are created for the Categorical variable “Category” and all other numerical variables have been normalized using the preprocess() function to achieve the same order of magnitude.

To find the best k, the model is trained on normalized training data with k=1 to k=25.

From the below graph, it is observed that the optimal value of k should be considered as 5 because higher the value of K, it has more chances of losing local proximity and very small value of k will have an overfitting issue. So, with accuracy as 0.9854 and from k=7 onwards the accuracy is not changing very much. Thus,k = 5 is selected.

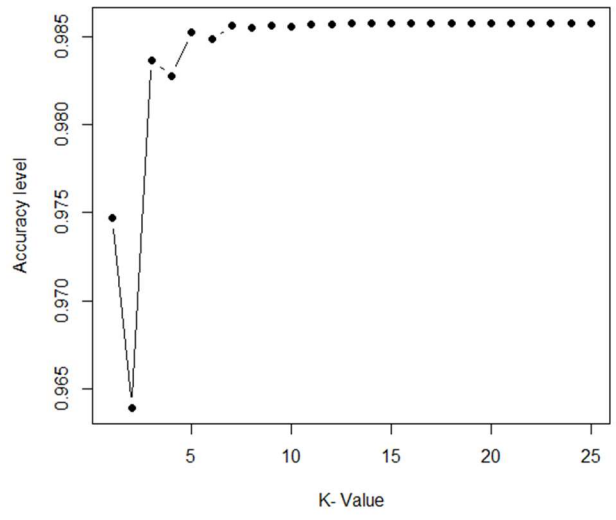


Fig 6.1.1.(a) k-values vs accuracy: decide optimal k value

```
> confusionMatrix(df.knn, as.factor(valid.df3$STATUS))
Confusion Matrix and Statistics

          Reference
Prediction APPROVED DENIED
  APPROVED   239672   3392
  DENIED        171     95

               Accuracy : 0.9854
                 95% CI : (0.9849, 0.9858)
    No Information Rate : 0.9857
    P-Value [Acc > NIR] : 0.9037

                  Kappa : 0.0487

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.99929
            Specificity : 0.02724
         Pos Pred Value : 0.98604
         Neg Pred Value : 0.35714
             Prevalence : 0.98567
         Detection Rate : 0.98497
   Detection Prevalence : 0.99891
      Balanced Accuracy : 0.51327

       'Positive' Class : APPROVED
```

Fig 6.1.1(b): Confusion matrix for accuracy measures

Finally, the model is trained with k=5 and tested with validation data. The accuracy on validation data is 98.54%. As the model has selected the positive class as APPROVED cases, accuracy of predicting **APPROVED cases (Sensitivity) is 99.92%** but predicting **DENIED cases(Specificity) is very low 2.72%** only. This means that there are high chances

that the model will falsely select the DENIED case as APPROVED case. Hence, this model is not suitable for prediction and it is required to develop another model.

6.1.2. CLASSIFICATION TREE

The classification Tree is developed in 3 steps to classify the status as 1(Approve) and 0(Denied). The tree steps were:

- **Default Tree:** To explore and understand the Status classification without any specific control on the tree formation. The **Accuracy** for this default tree is **98.6%** with Specificity (Accuracy for Denied cases) as 0% and sensitivity(Accuracy for Approve cases) as **100%**.
- **Pruned Tree:** In the next Tree, the Default tree was pruned by finding the best cp with least x-error to get a better tree with improved accuracy.

For cp =0.00052, the **Accuracy** for Pruned tree is **98.6%** with Specificity (Accuracy for Denied cases) as 1% and Sensitivity(Accuracy for Approve cases) as **100%**. Refer below the Pruned Tree with Confusion Matrix and Statistics for the same.

	CP	nsplit	rel error	xerror	xstd
1	0.009409409	0	1.00000	1.00000	0.014052
2	0.000520521	1	0.99059	0.99059	0.013987
3	0.000266934	7	0.98739	0.99099	0.013989
4	0.000200200	12	0.98559	0.99279	0.014002
5	0.000133467	14	0.98519	0.99600	0.014024
6	0.000127400	17	0.98478	1.00100	0.014059
7	0.000120120	31	0.98278	1.00881	0.014113
8	0.000114400	46	0.98098	1.00881	0.014113
9	0.000100100	69	0.97818	1.01121	0.014130
10	0.000085800	86	0.97618	1.01341	0.014145
11	0.000080080	107	0.97437	1.02523	0.014226
12	0.000071500	131	0.97217	1.02523	0.014226
13	0.000070659	150	0.97077	1.03443	0.014289
14	0.000066733	197	0.96657	1.03443	0.014289
15	0.000063221	301	0.95896	1.03824	0.014314
16	0.000057200	337	0.95616	1.04585	0.014366
17	0.000050050	344	0.95576	1.04925	0.014389
18	0.000040040	422	0.95135	1.05345	0.014417
19	0.000037538	489	0.94815	1.06066	0.014466
20	0.000036400	514	0.94715	1.06226	0.014477
21	0.000033367	525	0.94675	1.06947	0.014525
22	0.000028600	608	0.94394	1.07367	0.014553
23	0.000026693	633	0.94314	1.07528	0.014564
24	0.000025025	663	0.94234	1.07568	0.014566
25	0.000023553	708	0.94114	1.07808	0.014582
26	0.000022244	729	0.94054	1.07928	0.014590
27	0.000018200	774	0.93954	1.08509	0.014629

Fig 6.2.1. Optimal value of cp selection

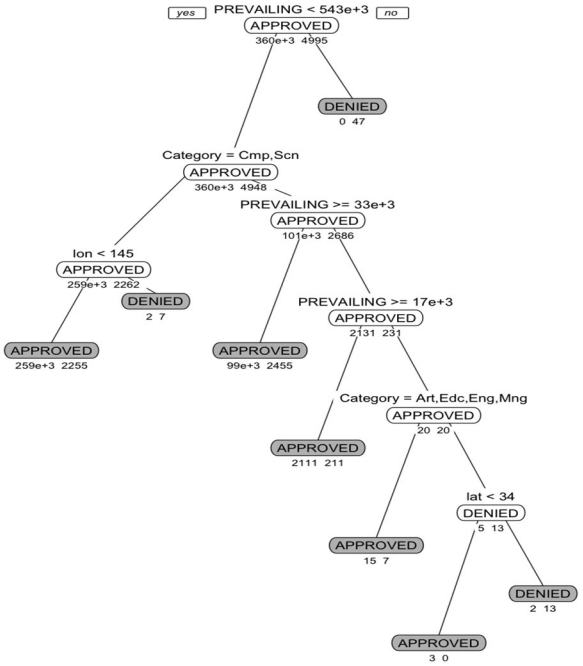


Fig 6.2.2. Pruned Tree with cp = 0.00052

```
Confusion Matrix and Statistics

          Reference
Prediction APPROVED DENIED
  APPROVED   239841   3449
  DENIED          2     38

               Accuracy : 0.9858
                 95% CI : (0.9853, 0.9863)
    No Information Rate : 0.9857
    P-Value [Acc > NIR] : 0.273

                  Kappa : 0.0212

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 1.0000
            Specificity : 0.0109
         Pos Pred Value : 0.9858
         Neg Pred Value : 0.9500
             Prevalence : 0.9857
         Detection Rate : 0.9857
   Detection Prevalence : 0.9998
      Balanced Accuracy : 0.5054

       'Positive' Class : APPROVED
```

Fig 6.2.3 accuracy for pruned classification tree

- **Random Forest:** After the Pruned tree, Random Forest is developed for 100 trees to find the predictors that contribute more to the classification of Status and Variable Importance Plot is made from Plot, it can be seen that **Category is the highest contributor** and appears in the maximum of trees for the classification(Refer below).

The **Accuracy** for Random Forest is **98.6%** with Specificity (Accuracy for Denied cases) as **3%** and Sensitivity(Accuracy for Approve cases) as **99.94%**.

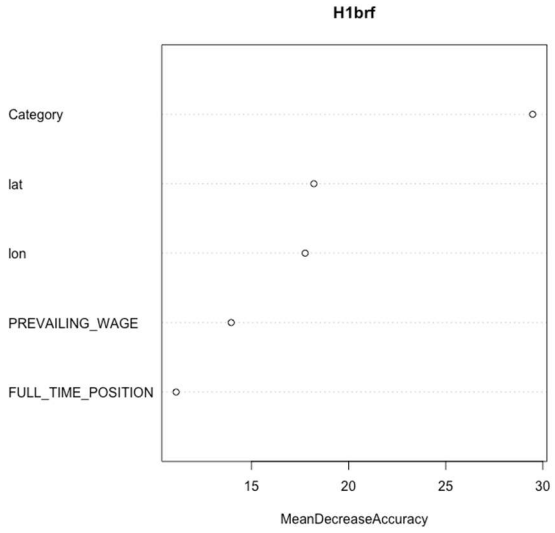


Fig 6.1.2(a)Variable Importance Plot by Random

```
> confusionMatrix(H1brf.pred.valid, as.factor(valid.df4$STATUS))
Confusion Matrix and Statistics

          Reference
Prediction APPROVED DENIED
  APPROVED   239701   3386
  DENIED        142    101

               Accuracy : 0.9855
                 95% CI : (0.985, 0.986)
    No Information Rate : 0.9857
    P-Value [Acc > NIR] : 0.7609

                  Kappa : 0.0524

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.99941
            Specificity : 0.02896
         Pos Pred Value : 0.98607
         Neg Pred Value : 0.41564
             Prevalence : 0.98567
         Detection Rate : 0.98509
   Detection Prevalence : 0.99900
      Balanced Accuracy : 0.51419

       'Positive' Class : APPROVED
```

Fig 6.1.2(b) Random Forest

The Accuracy, Sensitivity and Specificity is similar for all the three steps with marginal improvement in Sensitivity (Accuracy for Denied cases) for Pruned Tree and Random Forest. Considering business application, where visa application is filed with application fees and for every denial status its a loss. As the Classification Tree is very low on accuracy (3%) for classification of denial cases, some better models shall be developed with higher accuracy for Denial Cases classification.

6.1.3. LOGISTIC REGRESSION

The variables “Category” and “Status” are changed to class factor before models are created. The result of the logistic regression model (Fig 6.1.3(a)) shows that the model will correctly predict the **approved cases** for **99.99% (specificity)**of the time but will perform quite miserably for the **denied case** with only **1% (sensitivity)** time being correct. **Total accuracy is 98.58%** for the Model. The analysis is done by taking the probability as 0.9 which means that the model will say a case will be approved only when it is 90% sure (an effort to deal with the bias-ness of the dataset).

```
> confusionMatrix(as.factor(ifelse(model1.pred > 0.9,1,0)), as.factor(valid.df2[,5]))
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0     35      7
         1   3452 239836

               Accuracy : 0.9858
                 95% CI : (0.9853, 0.9863)
    No Information Rate : 0.9857
    P-Value [Acc > NIR] : 0.3203

                  Kappa : 0.0195

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.0100373
            Specificity : 0.9999708
         Pos Pred Value : 0.8333333
         Neg Pred Value : 0.9858111
             Prevalence : 0.0143303
         Detection Rate : 0.0001438
   Detection Prevalence : 0.0001726
      Balanced Accuracy : 0.5050040

       'Positive' Class : 0
```

Fig 6.1.3(a) Confusion Matrix for the Logistic model.

From the summary of the logistic model(Fig 6.1.3(b)), it is clear that the probability of getting an H1B application approved is lowest in the category "**Other**" as the Coefficient for "**Other**" is negative(**-0.0063**). The probability is also showing a negative relationship with "PREVAILING_WAGE" because there are a lot of outliers in PREVAILING_WAGE with a lot of extreme high wages being denied. Also the probability of getting H1B increases as the latitude increases and increases as the longitude decreases, thus indicating the region where most of the H1B applications are approved.

```
> summary(model1)

Call:
glm(formula = STATUS ~ ., family = binomial(link = logit), data = train.df2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.2021   0.1267   0.1342   0.1888   0.4980

Coefficients:
                      Estimate    Std. Error z value             Pr(>|z|)
(Intercept)          2.2247443441  0.1410049602  15.778 < 0.0000000000000002 ***
FULL_TIME_POSITION   0.0929636087  0.0376251299   2.471              0.01348 *
PREVAILING_WAGE     -0.0000022586  0.0000005161  -4.377    0.000012059815439 ***
lon                 -0.0013100939  0.0006757112  -1.939              0.05252 .
lat                  0.0271424772  0.0029705361   9.137 < 0.0000000000000002 ***
CategoryBusiness     0.5629879855  0.0756892399   7.438    0.000000000000102 ***
CategoryComputer     1.5111334477  0.0563700188  26.807 < 0.0000000000000002 ***
CategoryEducation    0.5088248554  0.1008014306   5.048    0.000000446940706 ***
CategoryEngineering  0.6870837903  0.0733197960   9.371 < 0.0000000000000002 ***
CategoryHealthcare   0.2728243727  0.0857377489   3.182              0.00146 **
CategoryManagement   0.6520271089  0.0701944399   9.289 < 0.0000000000000002 ***
CategoryOther       -0.0063089523  0.0644478560  -0.098              0.92202
CategoryScience      1.0740747882  0.0954338748  11.255 < 0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 52793  on 364993  degrees of freedom
Residual deviance: 50657  on 364981  degrees of freedom
AIC: 50683

Number of Fisher Scoring iterations: 11
```

Fig 6.1.3.(b) Summary of the Logistic Regression model showing coefficient values(estimate) for different Predictors

6.1.4. NEURAL NETWORK

Dummies are created for “Categories” and “STATUS” then in pre-processing the scaling of the continuous variable: “PREVAILING_WAGE”, “lon”, “lat”, are done before creating the models. Neural Net Model can be seen in Fig 6.1.4(a). Hidden layer 1 with 3 nodes is used in producing the NN model.

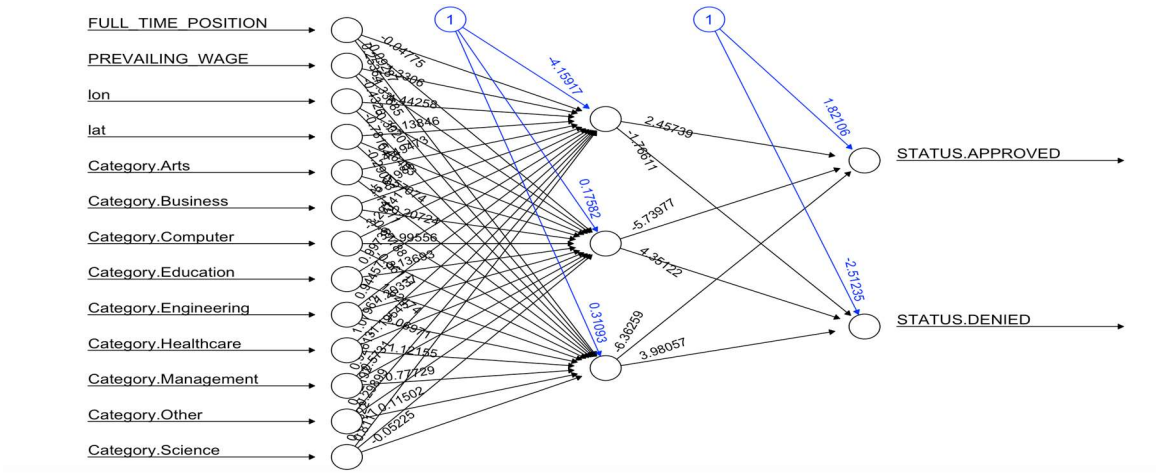


Fig 6.1.4(a) Plot of the Neural Net Model

The neural net model (Fig 6.1.4(b)) shows that it is doing comparatively well in predicting the **denied cases** with the performance of **13.96%(sensitivity)**, which is remarkably higher than what other Models have performed. Performance for **approved cases** is **98.59% (specificity)** and the total **accuracy** of the model is **98.41%**. The analysis is done by taking the probability as 0.9 which means that the model will say a case will be approved only when it is 90% sure (an effort to deal with the bias-ness of the dataset).

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0     74   3413
         1    456 239387

               Accuracy : 0.9841
                 95% CI : (0.9836, 0.9846)
    No Information Rate : 0.9978
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0332

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.1396226
            Specificity : 0.9859432
         Pos Pred Value : 0.0212217
         Neg Pred Value : 0.9980988
             Prevalence : 0.0021781
         Detection Rate : 0.0003041
   Detection Prevalence : 0.0143303
      Balanced Accuracy : 0.5627829

       'Positive' Class : 0
```

Fig 6.1.4(b) Confusion Matrix for the Neural Net

6.2 MODELS WITH UNDERSAMPLED DATASET

New models were created with undersampling to account for the imbalance in the dataset. To do this, a high number of Approved cases were left out of the training set to create

a more even ratio between the original **98.7%** Approved and **1.3%** Denied cases. It was hoped that this new model would perform better with real world data than the previous models were capable of. The new dataset dimension changed to **28,304** records with **30%** of the data consisting of denied cases.

6.2.1. KNN

After pre-processing, the model was developed with the undersampled dataset using k=19 (highest accuracy and taking local structure in consideration)(Fig 6.2.1(a)).

The chances of correctly predicting the denied case by using the KNN algorithm with the undersampled dataset is increased to **27.67%** . Although the overall accuracy is reduced to **71.82%** and the performance for approved cases is reduced to **91.11%** but still the model is quite realistic from the perspective of denied cases (Fig 6.2.1(b)).

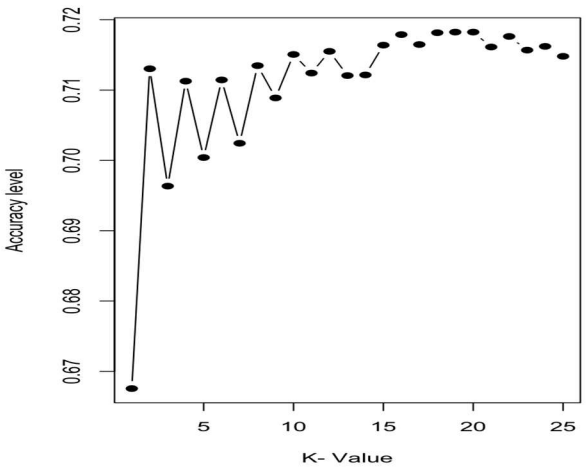


Fig 6.2.1(a) The graph for K-value vs accuracy of the

```
> confusionMatrix(df.knnunder, as.factor(valid.(
Confusion Matrix and Statistics

          Reference
Prediction APPROVED DENIED
  APPROVED     7180   2489
  DENIED        701    952

               Accuracy : 0.7182
                 95% CI : (0.7099, 0.7265)
    No Information Rate : 0.6961
    P-Value [Acc > NIR] : 0.0000001277

                  Kappa : 0.2199

 Mcnemar's Test P-Value : < 0.00000000000000022

            Sensitivity : 0.9111
            Specificity : 0.2767
         Pos Pred Value : 0.7426
         Neg Pred Value : 0.5759
             Prevalence : 0.6961
         Detection Rate : 0.6342
   Detection Prevalence : 0.8540
      Balanced Accuracy : 0.5939

       'Positive' Class : APPROVED
```

Fig 6.2.1(c) Confusion matrix for the KNN

6.2.2. CLASSIFICATION TREE

The pruned tree looks very specific to come up with different rules for the status classification. For example:

IF “Category” <> “Computers” or “Science” AND “PREVILING_WAGE” < 40000 THEN Class(DENIED).

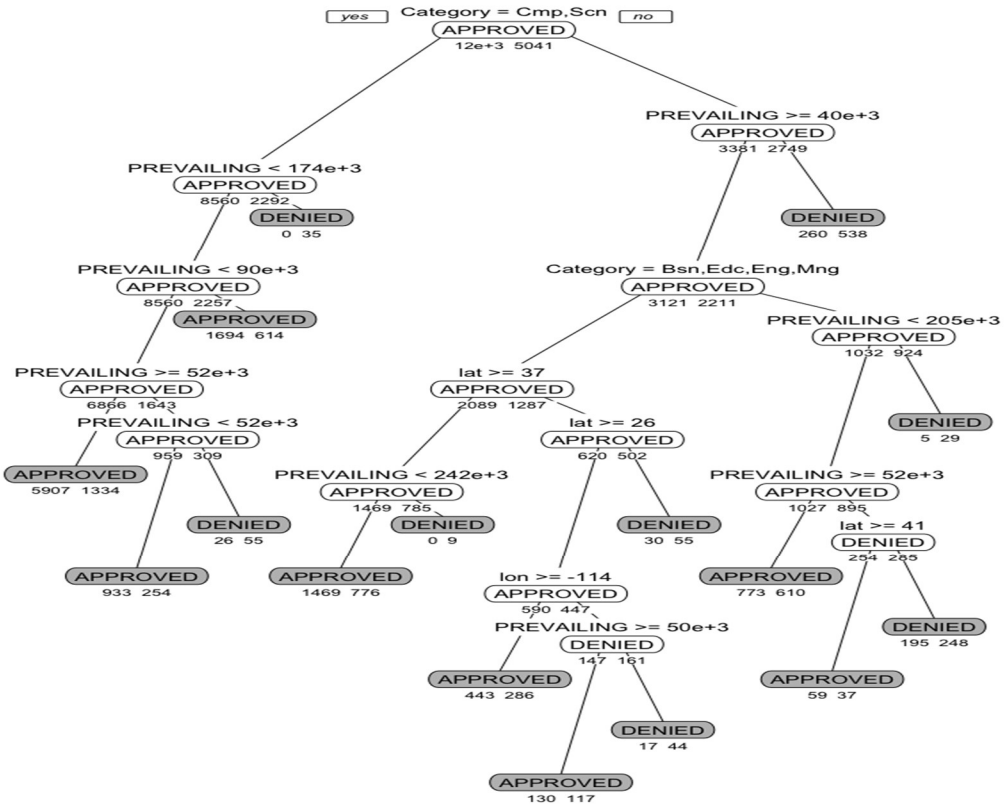


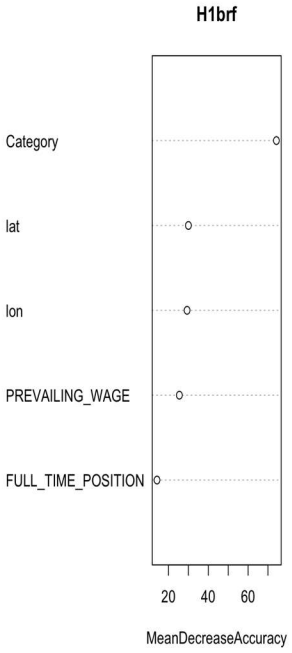
Fig 6.2.2(a) Pruned Classification Tree

```
> confusionMatrix(H1bpruned.pred.valid, as.factor(valid.df4under$STATUS))
Confusion Matrix and Statistics

          Reference
Prediction APPROVED DENIED
  APPROVED      7505   2790
  DENIED         376    651

               Accuracy : 0.7204
                 95% CI : (0.712, 0.7286)
    No Information Rate : 0.6961
    P-Value [Acc > NIR] : 0.000000007948

                  Kappa : 0.1763

 Mcnemar's Test P-Value : < 0.00000000000000022

            Sensitivity : 0.9523
            Specificity : 0.1892
         Pos Pred Value : 0.7290
         Neg Pred Value : 0.6339
             Prevalence : 0.6961
         Detection Rate : 0.6629
   Detection Prevalence : 0.9093
      Balanced Accuracy : 0.5707

       'Positive' Class : APPROVED
```

Fig 6.2.2(b) Confusion matrix for Pruned Classification Tree

Random Forest:

With the undersampled data, Random Forest Fig 6.2.2(d) shows an improvement in classifying the denied cases to **31.3%** as compared to earlier models. The overall model accuracy is **72.4%** with prediction accuracy of approved cases as **95.23%**. With these accuracy measures, it can be concluded that there is considerable improvement in model applicability to the real world.




```
> confusionMatrix(H1brf.pred.valid, as.factor(valid.df4under$STATUS))
Confusion Matrix and Statistics

          Reference
Prediction APPROVED DENIED
  APPROVED     7255   2362
  DENIED        626   1079

               Accuracy : 0.7361
                 95% CI : (0.7279, 0.7442)
    No Information Rate : 0.6961
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 0.2729

 Mcnemar's Test P-Value : < 0.00000000000000022

            Sensitivity : 0.9206
            Specificity : 0.3136
         Pos Pred Value : 0.7544
         Neg Pred Value : 0.6328
             Prevalence : 0.6961
         Detection Rate : 0.6408
   Detection Prevalence : 0.8494
      Balanced Accuracy : 0.6171

       'Positive' Class : APPROVED
```

Fig 6.2.2(c) Variable Importance Plot by Random Forest; (d) Random Forest Confusion Matrix

6.2.3. LOGISTIC REGRESSION

In the logistic regression model the accuracy of predicting the denied cases is increased to **24.38%** versus ~ **1%** for the original dataset model(Fig 6.2.3(a)). The overall model accuracy is **71.16%** and approval cases prediction accuracy is **91.59%** which is relatively good for a model applicability.

```
> confusionMatrix(as.factor(ifelse(model1.predunder > 0.5,1,0)), as.factor(valid.df2under[,5]))
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0  839  663
         1 2602 7218

               Accuracy : 0.7116
                 95% CI : (0.7032, 0.72)
    No Information Rate : 0.6961
    P-Value [Acc > NIR] : 0.0001577

                  Kappa : 0.1898

 Mcnemar's Test P-Value : < 0.00000000000000022

            Sensitivity : 0.2438
            Specificity : 0.9159
         Pos Pred Value : 0.5586
         Neg Pred Value : 0.7350
             Prevalence : 0.3039
         Detection Rate : 0.0741
   Detection Prevalence : 0.1327
      Balanced Accuracy : 0.5798

       'Positive' Class : 0
```

Fig 6.2.3(a) Confusion matrix for Logistic Regression

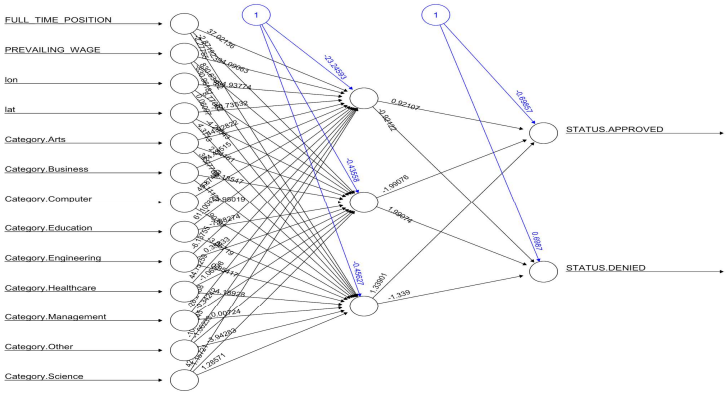
The relation between the predictors and the probability of getting an H1B application getting approved remained the same(based on the nature of coefficients of the predictors)(Fig 6.2.3(b)).

```
> summary(model1under)

Call:
glm(formula = STATUS ~ ., family = binomial(link = logit), data = train.df2under)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8801  -1.1447   0.6715   0.7259   1.4586

Coefficients:
                       Estimate    Std. Error z value             Pr(>|z|)
(Intercept)         -0.7642955200  0.1701933615  -4.491      0.0000070973439 ***
FULL_TIME_POSITION   0.0752011753  0.0435839384   1.725             0.084449 .
PREVAILING_WAGE     -0.0000019149  0.0000005433  -3.524             0.000425 ***
lon                 -0.0003084117  0.0007821333  -0.394             0.693344
lat                  0.0188091310  0.0036186764   5.198      0.0000002016685 ***
CategoryBusiness     0.5003140894  0.0997993373   5.013      0.0000005353206 ***
CategoryComputer     1.4720923831  0.0776433102  18.960 < 0.0000000000000002 ***
CategoryEducation    0.5717128387  0.1332684852   4.290      0.0000178727235 ***
CategoryEngineering  0.6453453887  0.0973253946   6.631      0.0000000000334 ***
CategoryHealthcare   0.1981454333  0.1175049048   1.686             0.091743 .
CategoryManagement   0.5823133827  0.0944436954   6.166      0.0000000007016 ***
CategoryOther       -0.0121069403  0.0918710333  -0.132             0.895157
CategoryScience      1.1593678388  0.1201367075   9.650 < 0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 20656  on 16981  degrees of freedom
Residual deviance: 19372  on 16969  degrees of freedom
AIC: 19398

Number of Fisher Scoring iterations: 11
```

Fig 6.2.3(b)Summary of the Logistic Regression model showing coefficient values(estimate) for different Predictors

6.2.4. NEURAL NETWORK

With the neural network method, the accuracy of predicting the denied cases is also remarkably increased by four times from **13.96%** with original dataset to **57.86%** with undersampling technique. Hidden layer 1 with 3 nodes is used in producing the NN model.




```
> confusionMatrix(as.factor(valid.df1under$STATUS.APPROVED)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0  812 2629
         1  601 7280

               Accuracy : 0.7147
                 95% CI : (0.7063, 0.723)
    No Information Rate : 0.8752
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1915

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.57466
            Specificity : 0.73469
         Pos Pred Value : 0.23598
         Neg Pred Value : 0.92374
             Prevalence : 0.12480
         Detection Rate : 0.07172
   Detection Prevalence : 0.30392
      Balanced Accuracy : 0.65467

       'Positive' Class : 0
```

Fig 6.2.4(b) Confusion Matrix for the Neural Net

7. COMPARISON OF MODELS

After implementing selected algorithms to the original and undersampled dataset, different models were created. The comparison of those models based on accuracy measures (for validation data) are presented in table 7.1.

	Accuracy Measures	KNN K=5	Classification Tree (Random Forest)	Logistic Regression	Neural Net Nodes = 3
With Normal	Model Accuracy%	98.54	98.55	98.58	**98.41**
	Chances of Correctly Predicting a Approved case(%)	99.92	99.94	99.99	**98.59**
	Chances of Correctly Predicting a Denied case(%)	2.72	2.9	1	**13.96**
	Accuracy Measures	**KNN K=19**	**Classification Tree (Random Forest)**	**Logistic Regression**	**Neural Net Nodes = 3**
With Under-	Model Accuracy%	71.82	73.61	71.16	**71.47**
	Chances of Correctly Predicting a Approved case(%)	91.11	92.06	91.59	**73.47**
	Chances of Correctly Predicting a Denied case(%)	27.67	31.36	24.38	**57.47**

Table 7.1: Comparison of Models

This project's primary purpose is to correctly predict denied cases. The penalty for failing at that task will be a financial one for the applicant companies, the applicant workers, and our own consultancy firm.

Comparing the result of the Models produced:

1. Based on the original dataset:

From Table 7.1, it is clear that the neural net model is the best for predicting the denied case(**13.96%**). Overall model accuracy (**98%**) and it's accuracy for predicting the approved case(**98.59%**) is comparable to all the other models using the original dataset.

2. Based on the undersampled dataset:

Using the undersampled dataset, the neural network model bests all other models with an achieved accuracy of **57.47%**. This is despite the fact that the NN achieves the lowest accuracy for predicting approved cases(**73.47%**). However, because the primary objective is to correctly predict denied cases, this is the best model to be used in real time to predict both the Case Status outcomes.

3. Comparing selected models from the Original dataset and the undersampled dataset:

From the findings above, this report can conclude that the undersampled neural net model has a lower general prediction accuracy, but higher prediction accuracy for denied cases. Because the main objective is to accurately predict denied cases, the undersampled neural network is the best model for real world applications.

8. CONCLUSION AND RECOMMENDATION

While this dataset is a rich environment for exploratory analysis, it challenges many of the traditional algorithms for prediction. However, once undersampling was performed, a comparison of accuracies proves that the method grants higher prediction quality for Denied cases. Undersampling also allows this report to claim that a more balanced dataset would have allowed greater overall accuracy than an imbalanced dataset.

A company that is able to spare clients the time and cost of failed applications would bring in a lot of customers. That business could use this report's model to aid in consulting clients with their H1-b applications, using it to quickly and accurately predict the outcome of their current application, and revising it to have a higher chance of being accepted.

FUTURE WORK

If granted more time, additional models could be devised from the existing data. For example, early trials with variables derived from existing data allowed for overall prediction accuracy approaching 99%, with correct prediction of denied cases at nearly 60% and correct prediction of accepted cases at over 99%. Using these techniques and expanding their functionality could have led to far better results than what the project's time frame allowed for.

Furthermore, clustering algorithms could be performed on the variables State or Industry, and then the analysis can be done on different clusters separately. By using clustering and undersampling together, significantly more accurate results could be achieved.

REFERENCES

1. Galit Shmueli, Peter C. Bruce, Inbal Yahav, Nitin R. Patel, Kenneth C. Lichtendhl Jr. Data Mining for Business Analytics Edition 2018
2. H. Wickham. **ggplot2**: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016
3. https://www.rdocumentation.org/
4. R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.
5. Nicola Lunardon, Giovanna Menardi, and Nicola Torelli (2014). **ROSE**: a Package for Binary Imbalanced Learning. R Journal, 6(1), 82-92.
6. Dean Attali and Christopher Baker (2019). **ggExtra**: Add Marginal Histograms to 'ggplot2', and More 'ggplot2' Enhancements. R package version 0.9. https://CRAN.R-project.org/package=ggExtra
7. Bob Rudis (2020). **hrbrthemes**: Additional Themes, Theme Components and Utilities for 'ggplot2'. R package version 0.8.0. https://CRAN.R-project.org/package=hrbrthemes
8. www.google.com