R Notebook

TOPIC - H1B Data Anlayis and Prediction

Summary H-1B visas are a category of employment-based, non-immigrant visas for temporary foreign workers in the United States. For a foreign national to apply for H1-B visa, a US employer must offer them a job and submit a petition for a H-1B visa to the US immigration department. The Reason I finalized this topic was because there have been a lot hoopla and misconceptions surrounding H1b visa and wanted to explore this domain in order to get the exact mathematical results in orders to test the feasibility of various information provided on online platform and understand what factors affect the H1b visas and also what are the best jobs and companies that get the most H1b sponsered.H1b dataset contains 6 years of datset from 2011 to 2016. I acquired the dataset from Office of Foreign Labor Certification (OFLC) websitethe following source:
https://www.foreignlaborcert.doleta.gov/performancedata.cfm.I
(https://www.foreignlaborcert.doleta.gov/performancedata.cfm.I) started with the Exploratry data analysis whih included detecting missing values, Visualization of data based on various parameters such as salary distribution ,top employers, jobs with most opportubities etc. I further continued with feature engineering and data cleaning and finally buit classification model for prediction.

```
library(plyr)
library(tibble)
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:plyr':
##
##     here
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(DT)
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday,
##     week, yday, year
```

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------
--------------- tidyverse 1.2.1 --
```

```
## v tidyr   1.0.0     v dplyr   0.8.3
## v readr   1.3.1     v stringr 1.4.0
## v purrr   0.3.3     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------------------
--------- tidyverse_conflicts() --
## x dplyr::arrange()         masks plyr::arrange()
## x lubridate::as.difftime() masks base::as.difftime()
## x dplyr::between()         masks data.table::between()
## x purrr::compact()         masks plyr::compact()
## x dplyr::count()           masks plyr::count()
## x lubridate::date()        masks base::date()
## x dplyr::failwith()        masks plyr::failwith()
## x dplyr::filter()          masks stats::filter()
## x dplyr::first()           masks data.table::first()
## x lubridate::here()        masks plyr::here()
## x data.table::hour()       masks lubridate::hour()
## x dplyr::id()              masks plyr::id()
## x lubridate::intersect()   masks base::intersect()
## x data.table::isoweek()    masks lubridate::isoweek()
## x dplyr::lag()             masks stats::lag()
## x dplyr::last()            masks data.table::last()
## x data.table::mday()       masks lubridate::mday()
## x data.table::minute()     masks lubridate::minute()
## x data.table::month()      masks lubridate::month()
## x dplyr::mutate()          masks plyr::mutate()
## x data.table::quarter()    masks lubridate::quarter()
## x dplyr::rename()          masks plyr::rename()
## x data.table::second()     masks lubridate::second()
## x lubridate::setdiff()     masks base::setdiff()
## x dplyr::summarise()       masks plyr::summarise()
## x dplyr::summarize()       masks plyr::summarize()
## x purrr::transpose()       masks data.table::transpose()
## x lubridate::union()       masks base::union()
## x data.table::wday()       masks lubridate::wday()
## x data.table::week()       masks lubridate::week()
## x data.table::yday()       masks lubridate::yday()
## x data.table::year()       masks lubridate::year()
```

```
library(pacman)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

Running the required library

```
h1b<-fread('h1b.csv')
```

We read the dataset using Fread command which helps you read the data set quickly instead of using read.csv
Exploratory Data Analysis

```
head(h1b)
```

	…	CASE_STATUS	EMPLOYER_NAME
		<int><chr>	<chr>
1		CERTIFIED-WITHDRAWN	UNIVERSITY OF MICHIGAN
2		CERTIFIED-WITHDRAWN	GOODMAN NETWORKS, INC.
3		CERTIFIED-WITHDRAWN	PORTS AMERICA GROUP, INC.
4		CERTIFIED-WITHDRAWN	GATES CORPORATION, A WHOLLY-OWNED SUBSIDIARY OF TOMKINS PLC
5		WITHDRAWN	PEABODY INVESTMENTS CORP.
6		CERTIFIED-WITHDRAWN	BURGER KING CORPORATION

6 rows | 1-3 of 11 columns

We see the overall dataset

```
str(h1b)
```

```
## Classes 'data.table' and 'data.frame':   3002458 obs. of  11 variables:
##  $ V1                : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ CASE_STATUS       : chr  "CERTIFIED-WITHDRAWN" "CERTIFIED-WITHDRAWN" "CERTIFIED-WITHDRAWN"
"CERTIFIED-WITHDRAWN" ...
##  $ EMPLOYER_NAME     : chr  "UNIVERSITY OF MICHIGAN" "GOODMAN NETWORKS, INC." "PORTS AMERICA
GROUP, INC." "GATES CORPORATION, A WHOLLY-OWNED SUBSIDIARY OF TOMKINS PLC" ...
##  $ SOC_NAME          : chr  "BIOCHEMISTS AND BIOPHYSICISTS" "CHIEF EXECUTIVES" "CHIEF EXECUTI
VES" "CHIEF EXECUTIVES" ...
##  $ JOB_TITLE         : chr  "POSTDOCTORAL RESEARCH FELLOW" "CHIEF OPERATING OFFICER" "CHIEF P
ROCESS OFFICER" "REGIONAL PRESIDEN, AMERICAS" ...
##  $ FULL_TIME_POSITION: chr  "N" "Y" "Y" "Y" ...
##  $ PREVAILING_WAGE   : num  36067 242674 193066 220314 157518 ...
##  $ YEAR              : int  2016 2016 2016 2016 2016 2016 2016 2016 2016 2016 ...
##  $ WORKSITE          : chr  "ANN ARBOR, MICHIGAN" "PLANO, TEXAS" "JERSEY CITY, NEW JERSEY" "D
ENVER, COLORADO" ...
##  $ lon               : num  -83.7 -96.7 -74.1 -105 -90.2 ...
##  $ lat               : num  42.3 33 40.7 39.7 38.6 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

We see that the dataset contains around 3 million observation and 11 variable in the data set. The dataset description is as follows:

CASE_STATUS - Status is associated with the last significant update. EMPLOYER_NAME - Name of the employers filing for petitions. SOC_NAME -Occupational name. JOB Title - Titles of different jobs FULL_TIME_POSITION - If the position is full time or not. PREVAILING_WAGE - average wage paid to similarly employed workers in the area of intended employment. YEAR -Year for filing the petition. WORKSITE - City and State information. Lon - Longitude of the worksite Lat - Latitude of the worksite

```
summary(h1b)
```

```
##        V1          CASE_STATUS        EMPLOYER_NAME
##  Min.   :      1   Length:3002458     Length:3002458
##  1st Qu.: 750615   Class :character   Class :character
##  Median :1501230   Mode  :character   Mode  :character
##  Mean   :1501230
##  3rd Qu.:2251844
##  Max.   :3002458
##
##    SOC_NAME          JOB_TITLE         FULL_TIME_POSITION
##  Length:3002458     Length:3002458     Length:3002458
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  PREVAILING_WAGE          YEAR        WORKSITE              lon
##  Min.   :0.000e+00   Min.   :2011   Length:3002458     Min.   :-157.86
##  1st Qu.:5.437e+04   1st Qu.:2012   Class :character   1st Qu.:-111.93
##  Median :6.502e+04   Median :2014   Mode  :character   Median : -86.16
##  Mean   :1.470e+05   Mean   :2014                      Mean   : -92.13
##  3rd Qu.:8.143e+04   3rd Qu.:2015                      3rd Qu.: -75.51
##  Max.   :6.998e+09   Max.   :2016                      Max.   : 145.73
##  NA's   :85          NA's   :13                        NA's   :107242
##       lat
##  Min.   :13.44
##  1st Qu.:34.17
##  Median :39.10
##  Mean   :38.16
##  3rd Qu.:40.88
##  Max.   :64.84
##  NA's   :107242
```

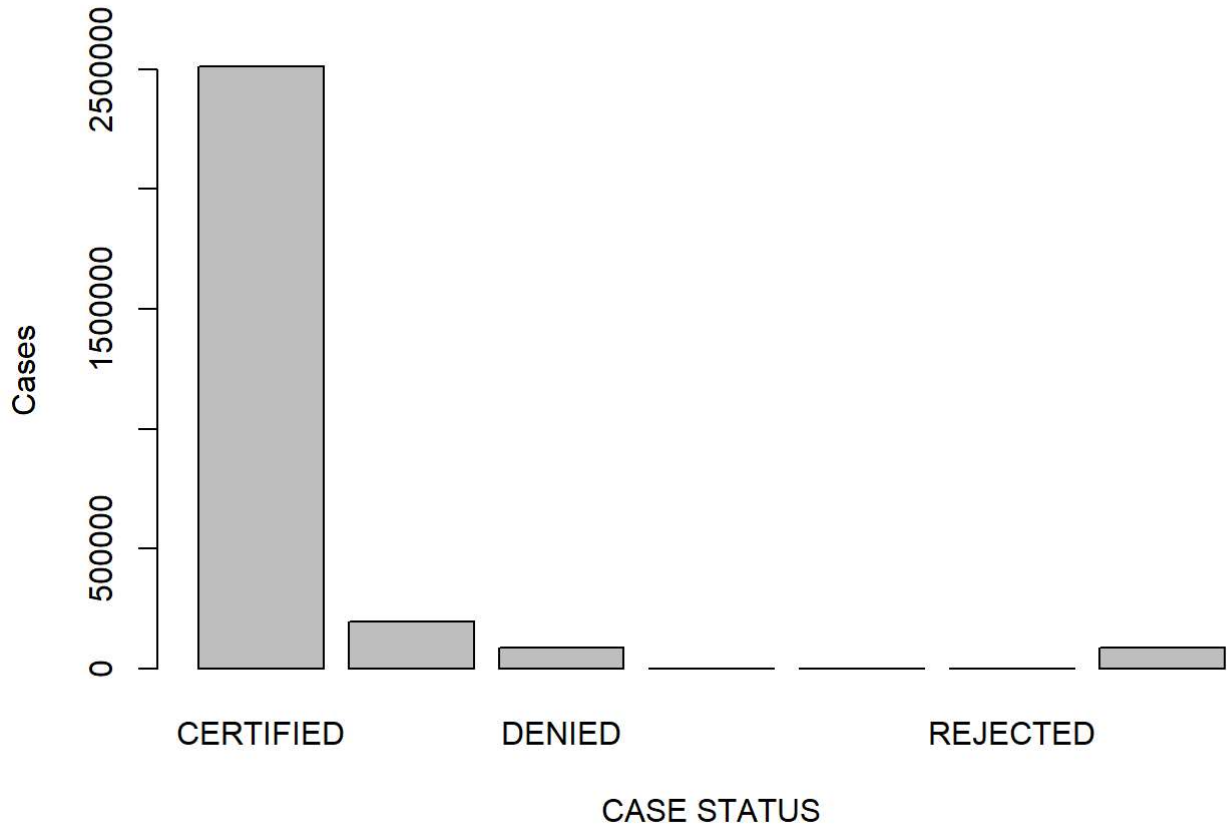
Here we observe that except year, prevailing wages, lattitude and longitutde evrything variable is character. Also lat and lon contains lot of nan values.

```
h1b <- na.omit(h1b)
summary(h1b)
```

```
##        V1          CASE_STATUS        EMPLOYER_NAME
##  Min.   :      1   Length:2877783     Length:2877783
##  1st Qu.: 740414   Class :character   Class :character
##  Median :1484054   Mode  :character   Mode  :character
##  Mean   :1489376
##  3rd Qu.:2236962
##  Max.   :3002445
##    SOC_NAME          JOB_TITLE         FULL_TIME_POSITION
##  Length:2877783     Length:2877783     Length:2877783
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##  PREVAILING_WAGE          YEAR        WORKSITE              lon
##  Min.   :0.000e+00   Min.   :2011   Length:2877783     Min.   :-157.86
##  1st Qu.:5.460e+04   1st Qu.:2012   Class :character   1st Qu.:-111.90
##  Median :6.512e+04   Median :2014   Mode  :character   Median : -86.16
##  Mean   :1.452e+05   Mean   :2014                      Mean   : -92.13
##  3rd Qu.:8.152e+04   3rd Qu.:2015                      3rd Qu.: -75.51
##  Max.   :6.998e+09   Max.   :2016                      Max.   : 145.73
##       lat
##  Min.   :13.44
##  1st Qu.:34.17
##  Median :39.10
##  Mean   :38.16
##  3rd Qu.:40.88
##  Max.   :64.84
```

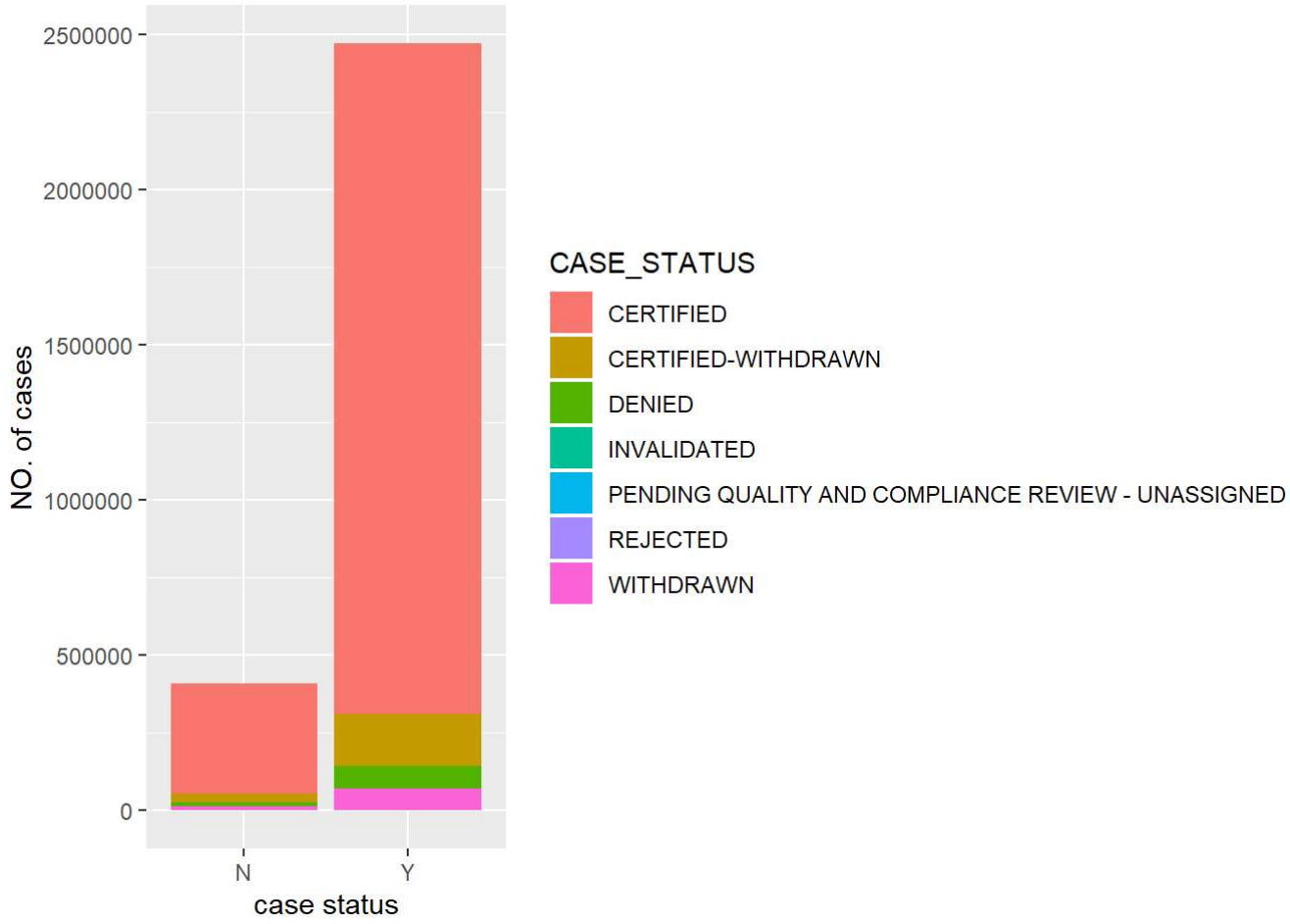
I decided to omit all the nan values as all the nan values were less than 10% of the no. of values of the variables

```
CASE_STATUS <- as.factor(h1b$CASE_STATUS)
plot(CASE_STATUS, xlab= "CASE STATUS" , ylab= 'Cases')
```



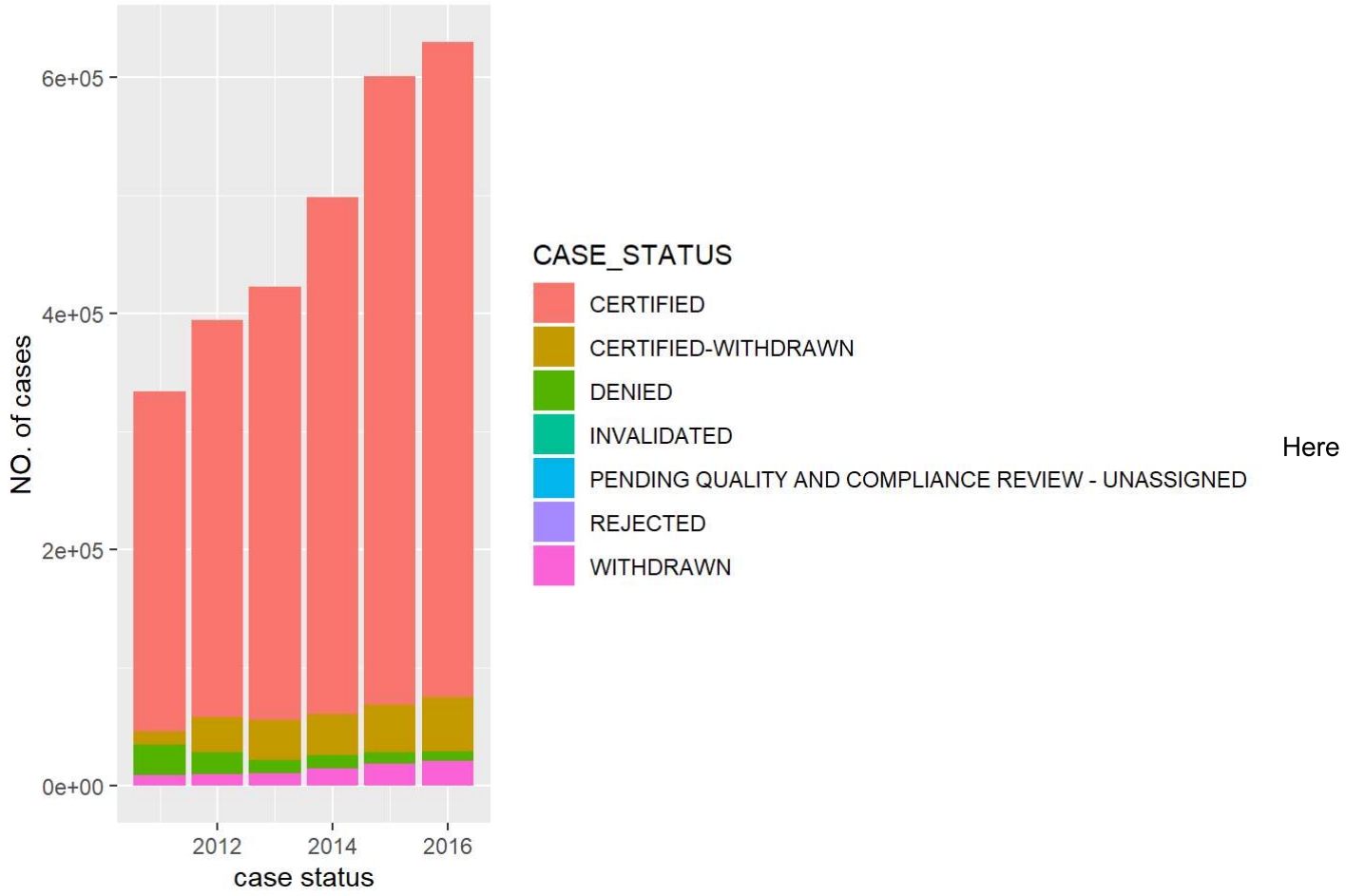
From above graph we can say that maximum of the classes belong to certified class through which we can conclude that the data is imbalance.This being our target variable reuires work to be done upon. During the featire enngineerg part we will look into this.

```
ggplot(h1b, aes(fill=CASE_STATUS, x= FULL_TIME_POSITION)) + geom_bar() +labs(y= 'NO. of cases' , x= 'case status')
```



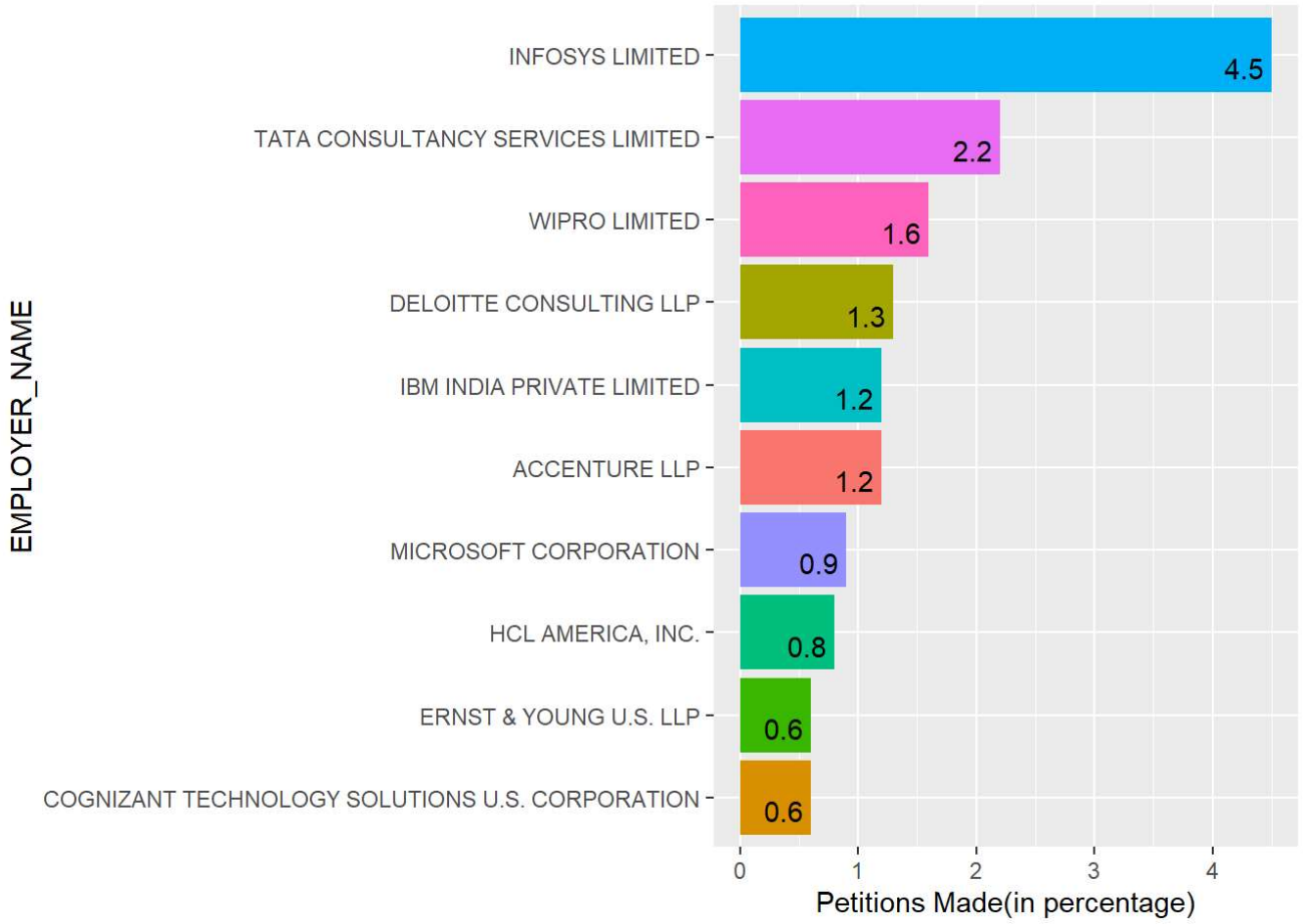
Here we see that maximum appicants lie in the full time zone and also maximum cases are certified

```
ggplot(h1b, aes(fill=CASE_STATUS, x= YEAR)) + geom_bar() +labs(y= 'NO. of cases' , x= 'case stat
us')
```



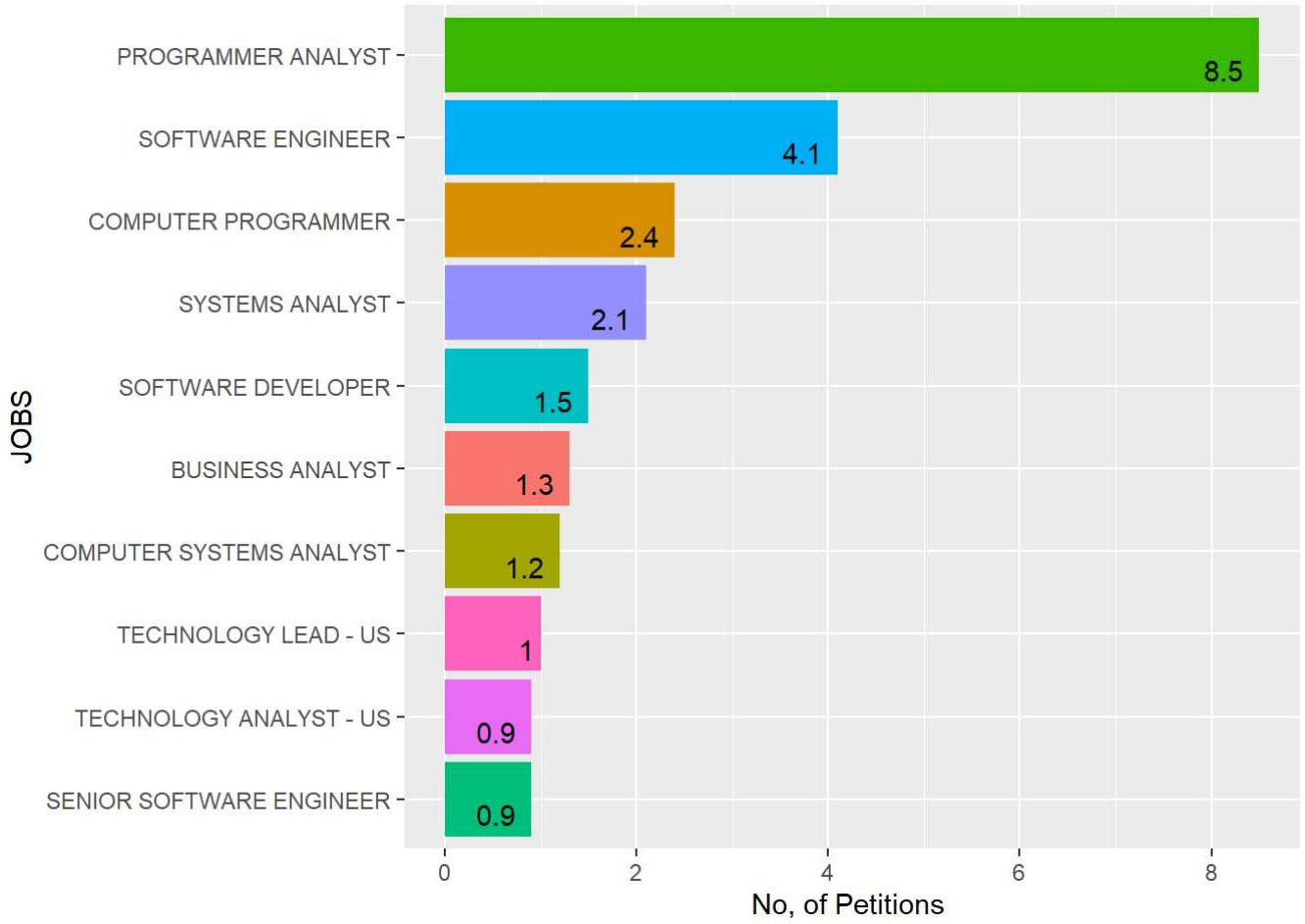
Here we see that as soon as the time increases the h1b visas also increse from 2011 to 2016

```
h1b$EMPLOYER_NAME <- factor(h1b$EMPLOYER_NAME)
Top_Sponser <- as.data.frame(h1b %>% group_by(EMPLOYER_NAME) %>%
                             summarise(count = n(), percent = round(count*100/nrow(h1b),1)) %>%
                             arrange(desc(count))%>%
                             top_n(10, wt = count))

ggplot(data = Top_Sponser, aes(x = reorder(EMPLOYER_NAME, percent),
                               y = percent, fill = EMPLOYER_NAME)) +
    geom_bar(stat = "identity") +
    geom_text(aes(label = percent), vjust = 1.1, hjust = 1.2) +
    labs(x = "EMPLOYER_NAME", y = "Petitions Made(in percentage)")  +
    theme(legend.position = "none") +
    coord_flip()
```



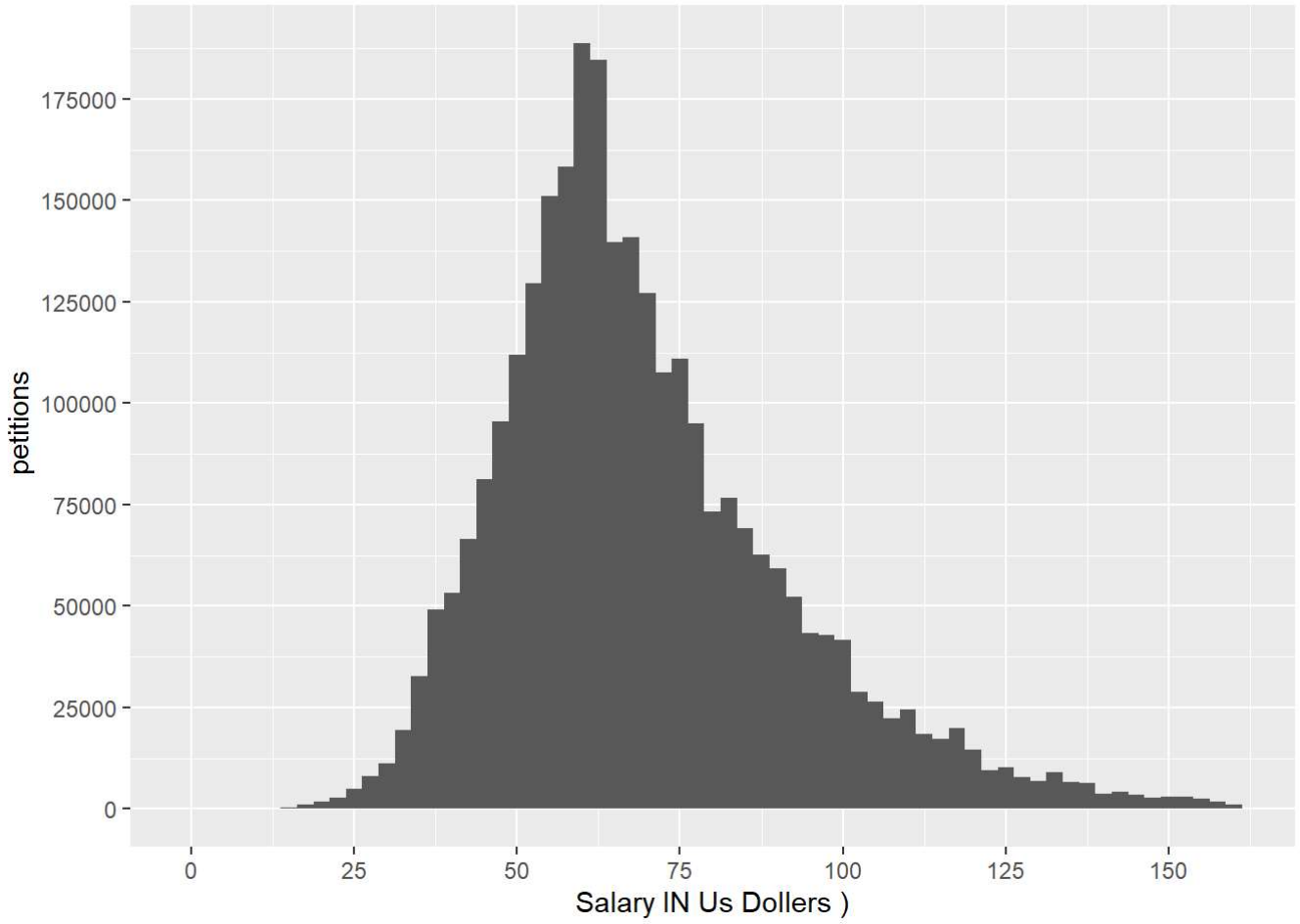
From the above graph we see that top companies that sponser h1b are indian companies getting into these companies incresaes the chance of getting h1b sponsered.

```
h1b$JOB_TITLE <- factor(h1b$JOB_TITLE)
best_jobs <- as.data.frame(h1b %>% group_by(JOB_TITLE) %>%
                             summarise(count = n(), percent = round(count*100/nrow(h1b),1)) %>%
                             arrange(desc(count))%>%
                             top_n(10, wt = count))

ggplot(data = best_jobs, aes(x = reorder(JOB_TITLE, percent),
                               y = percent, fill = JOB_TITLE)) +
    geom_bar(stat = "identity") +
    geom_text(aes(label = percent), vjust = 1.2, hjust = 1.4) +
    labs(x = "JOBS", y = "No, of Petitions ") +
    theme(legend.position = "none") +
    coord_flip()
```



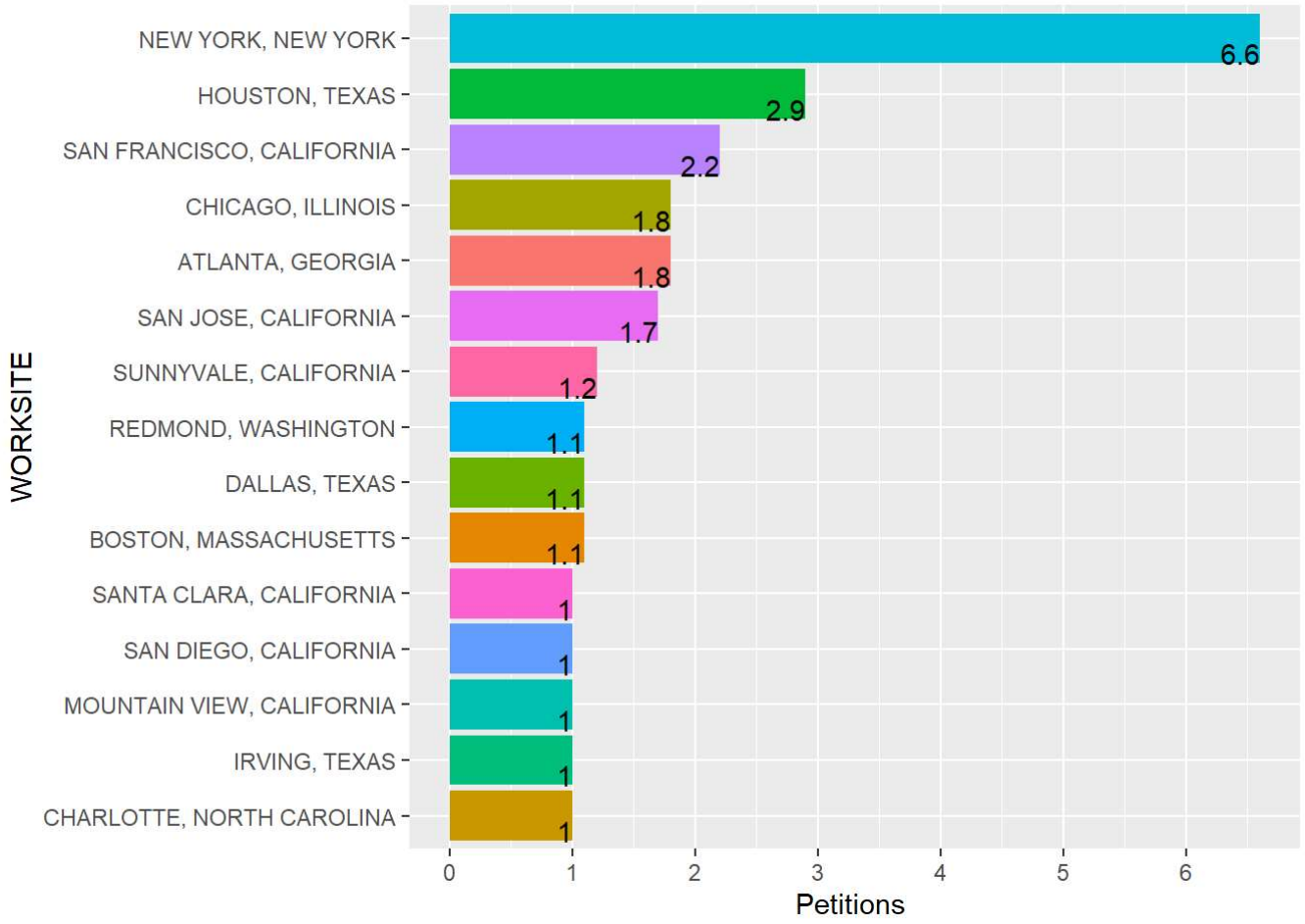
From the above graph we see that the best job of getting h1b sponsered are analyst roles therefore getting into this domain help you getting sponsered better than others

```
ggplot(data = subset(h1b, h1b$PREVAILING_WAGE < quantile(h1b$PREVAILING_WAGE, 0.99, na.rm = T)),
       aes(x = PREVAILING_WAGE/1000)) +
    geom_histogram(binwidth = 2.5) +
    scale_x_continuous(breaks = seq(0,150,25)) +
    scale_y_continuous(breaks = seq(0,500000,25000)) +
    labs(x = "Salary IN Us Dollers )", y = " petitions")
```



The above graph shows that maximum applicants lie between 50K to 75k zone . so Even Having H1b Sponsered does not gurranty good salary

```
h1b$WORKSITE <- factor(h1b$WORKSITE)
Best_location <- as.data.frame(h1b %>% group_by(WORKSITE) %>%
                summarise(count = n(), percent = round(count*100/nrow(h1b),1)) %>%
                arrange(desc(count))%>%
                top_n(15, wt = count))

ggplot(data = Best_location, aes(x = reorder(WORKSITE, percent),
                                 y = percent, fill = WORKSITE)) +
    geom_bar(stat = "identity") +
    geom_text(aes(label = percent), vjust = 1.2, hjust = 1.) +
    labs(x = "WORKSITE", y = "Petitions ") +
    scale_y_continuous(breaks = seq(0,7,1)) +
    theme(legend.position = "none") +
    coord_flip()
```



The above graph shows that califoria has the largest no. of people getiing h1b sponsered. Belonging to this state provides you a better opprtunity than others

```
chisq.test(h1b$CASE_STATUS, h1b$SOC_NAME, correct= F)
```

```
## Warning in chisq.test(h1b$CASE_STATUS, h1b$SOC_NAME, correct = F): Chi-
## squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  h1b$CASE_STATUS and h1b$SOC_NAME
## X-squared = 173484, df = 12258, p-value < 2.2e-16
```

When we run the chi square for SOC_NAME We see that it is significant as p value is less 1%

```
chisq.test(h1b$CASE_STATUS, h1b$YEAR, correct= F)
```

```
## Warning in chisq.test(h1b$CASE_STATUS, h1b$YEAR, correct = F): Chi-squared
## approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  h1b$CASE_STATUS and h1b$YEAR
## X-squared = 50033, df = 30, p-value < 2.2e-16
```

When we run the chi square for YEAR We see that it is significant as p value is less 1%

```
chisq.test(h1b$CASE_STATUS, h1b$JOB_TITLE, correct= F)
```

```
## Warning in chisq.test(h1b$CASE_STATUS, h1b$JOB_TITLE, correct = F): Chi-
## squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  h1b$CASE_STATUS and h1b$JOB_TITLE
## X-squared = 1681960, df = 1652886, p-value < 2.2e-16
```

When we run the chi square for JOB TITLE We see that it is significant as p value is less 1%

```
chisq.test(h1b$CASE_STATUS, h1b$FULL_TIME_POSITION, correct= F)
```

```
## Warning in chisq.test(h1b$CASE_STATUS, h1b$FULL_TIME_POSITION, correct =
## F): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  h1b$CASE_STATUS and h1b$FULL_TIME_POSITION
## X-squared = 553.61, df = 6, p-value < 2.2e-16
```

When we run the chi square for FULL TIME POSITION We see that it is significant as p value is less 1%

```
chisq.test(h1b$CASE_STATUS, h1b$EMPLOYER_NAME, correct= F)
```

```
## Warning in chisq.test(h1b$CASE_STATUS, h1b$EMPLOYER_NAME, correct = F):
## Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  h1b$CASE_STATUS and h1b$EMPLOYER_NAME
## X-squared = 3141287, df = 1289892, p-value < 2.2e-16
```

When we run the chi square for EMPLOYERS NAME We see that it is significant as p value is less 1%

Data Preparation

```
n<- dim(h1b)[1]
n
```

```
## [1] 2877783
```

```
h1b.data1 <- h1b[1:(n-2857783)]
h1b.data1
```

…	CASE_STATUS <int><chr>	EMPLOYER_NAME <fctr>
1	CERTIFIED-WITHDRAWN	UNIVERSITY OF MICHIGAN
2	CERTIFIED-WITHDRAWN	GOODMAN NETWORKS, INC.
3	CERTIFIED-WITHDRAWN	PORTS AMERICA GROUP, INC.
4	CERTIFIED-WITHDRAWN	GATES CORPORATION, A WHOLLY-OWNED SUBSIDIARY OF TOMKINS PLC
5	WITHDRAWN	PEABODY INVESTMENTS CORP.
6	CERTIFIED-WITHDRAWN	BURGER KING CORPORATION
7	CERTIFIED-WITHDRAWN	BT AND MK ENERGY AND COMMODITIES
8	CERTIFIED-WITHDRAWN	GLOBO MOBILE TECHNOLOGIES, INC.
10	WITHDRAWN	LESSARD INTERNATIONAL LLC
11	CERTIFIED-WITHDRAWN	H.J. HEINZ COMPANY

1-10 of 10,000 rows | 1-3 of 11 columns Previous **1** 2 3 4 5 6 … 1000 Next

After trying multiple times to run the classification model on 3 million roles i was unable to get the result so in order run the classfication model i cut down the dataset to 20000 rows to run the classification model.

There are lot of values in EMPLOYEE NAME , Even if we divide them into category it will not be feasable to use in the model there I decided to make a column containing top 5 companies, University name an otheres. From the Domain knowledge i got that If a person studies from the university in US he has a better chances of getting certified and sponsered therfore i decided to capture university.

```
h1b.data1$TOP_Sponsers  <- NA
```

We create a new column with nan values

```
h1b.data1$EMPLOYER_NAME <- tolower(h1b.data1$EMPLOYER_NAME)
```

in order to extract our required information we need to convert them into lower case letter. From the above code we did the reuired

```
h1b.data1$TOP_Sponsers[grep("University",h1b.data1$EMPLOYER_NAME, ignore.case = T)]<-"Universit
y"
```

We map the univeristy name into new column. All the strings that contain university are converted into common name “University”

```
h1b.data1$TOP_Sponsers[grep("infosys",h1b.data1$EMPLOYER_NAME, ignore.case = T)]<-"Top 5"
h1b.data1$TOP_Sponsers[grep("wipro",h1b.data1$EMPLOYER_NAME, ignore.case = T)]<-"Top 5"
h1b.data1$TOP_Sponsers[grep("ibm",h1b.data1$EMPLOYER_NAME, ignore.case = T)]<-"Top 5"
h1b.data1$TOP_Sponsers[grep("deloitte",h1b.data1$EMPLOYER_NAME, ignore.case = T)]<-" Top 5"
h1b.data1$TOP_Sponsers[grep("tata",h1b.data1$EMPLOYER_NAME, ignore.case = T)]<-"Top 5"
```

We map the top 5 company that sponser h1b into new column. All the strings that contain top 5 company name are converted into common name “Top 5”

```
h1b.data1$TOP_Sponsers[is.na(h1b.data1$TOP_Sponsers)] <- "others"
```

Rest of the nan values are conveted into others

SOC_Name Contains a lot of unique values there so i divided the column into group that of containg job of similar domain as mentioned below . For exam computer and database are grouptogther into cs/it. Similary math ans stats are grouped together into maths group.

```
h1b.data1$Imp_occupation <-NA
h1b.data1$SOC_NAME <- tolower(h1b.data1$SOC_NAME)
h1b.data1$Imp_occupation[grep('computer','programmer',h1b.data1$SOC_NAME, ignore.case = T)]<-"C
S/IT"
h1b.data1$Imp_occupation[grep('database',h1b.data1$SOC_NAME, ignore.case = T)]<-"CS/IT"
h1b.data1$Imp_occupation[grep('software','web developer',h1b.data1$SOC_NAME, ignore.case = T)]<-
"CS/IT"
h1b.data1$Imp_occupation[grep('math','statistic',h1b.data1$SOC_NAME, ignore.case = T)]<-"Maths"
h1b.data1$Imp_occupation[grep('predictive model','stats',h1b.data1$SOC_NAME, ignore.case = T)]<-
"Maths"
h1b.data1$Imp_occupation[grep('teacher','linguist',h1b.data1$SOC_NAME, ignore.case = T)]<-"Teach
er"
h1b.data1$Imp_occupation[grep('professor','Teach',h1b.data1$SOC_NAME, ignore.case = T)]<-"Taeche
r"
h1b.data1$Imp_occupation[grep('school principal',h1b.data1$SOC_NAME, ignore.case = T)]<-"Teache
r"
h1b.data1$Imp_occupation[grep('medical','doctor',h1b.data1$SOC_NAME, ignore.case = T)]<-"Medica
l"
h1b.data1$Imp_occupation[grep('physician','dentist',h1b.data1$SOC_NAME, ignore.case = T)]<-"Medi
cal"
h1b.data1$Imp_occupation[grep('Health','Physical Therapists',h1b.data1$SOC_NAME, ignore.case = T
)]<-"Medical"
h1b.data1$Imp_occupation[grep('medical','doctor',h1b.data1$SOC_NAME, ignore.case = T)]<-"Medica
l"
h1b.data1$Imp_occupation[grep('surgeon','nurse',h1b.data1$SOC_NAME, ignore.case = T)]<-"Medical"
h1b.data1$Imp_occupation[grep('psychiatr',h1b.data1$SOC_NAME, ignore.case = T)]<-"Medical"
h1b.data1$Imp_occupation[grep('chemist','physicist',h1b.data1$SOC_NAME, ignore.case = T)]<-"Scie
nce"
h1b.data1$Imp_occupation[grep('biology','scientist',h1b.data1$SOC_NAME, ignore.case = T)]<-"Scie
nce"
h1b.data1$Imp_occupation[grep('biology','clinical research',h1b.data1$SOC_NAME, ignore.case = T
)]<-"Science"
h1b.data1$Imp_occupation[grep('public relation','manager',h1b.data1$SOC_NAME, ignore.case = T)]<
-"Management"
h1b.data1$Imp_occupation[grep('management','operation',h1b.data1$SOC_NAME, ignore.case = T)]<-"M
anagement"
h1b.data1$Imp_occupation[grep('chief','plan',h1b.data1$SOC_NAME, ignore.case = T)]<-"Management"
h1b.data1$Imp_occupation[grep('executive',h1b.data1$SOC_NAME, ignore.case = T)]<-"Management"
h1b.data1$Imp_occupation[grep('promotion','market research',h1b.data1$SOC_NAME, ignore.case = T
)]<-"Marketing"
h1b.data1$Imp_occupation[grep('advertis','marketing',h1b.data1$SOC_NAME, ignore.case = T)]<-"Mar
keting"
h1b.data1$Imp_occupation[grep('business systems analyst',h1b.data1$SOC_NAME, ignore.case = T)]<-
"Business"
h1b.data1$Imp_occupation[grep('business','business analyst',h1b.data1$SOC_NAME, ignore.case = T
)]<-"Business"
h1b.data1$Imp_occupation[grep('accountant','finance',h1b.data1$SOC_NAME, ignore.case = T)]<-"Fin
ance"
h1b.data1$Imp_occupation[grep('engineer','architect',h1b.data1$SOC_NAME, ignore.case = T)]<-"Arc
hitecture"
h1b.data1$Imp_occupation[grep('surveyor','carto',h1b.data1$SOC_NAME, ignore.case = T)]<-"Archite
cture"
h1b.data1$Imp_occupation[grep('financial',h1b.data1$SOC_NAME, ignore.case = T)]<-"Finance"
```

```
h1b.data1$Imp_occupation[grep('technician','drafter',h1b.data1$SOC_NAME, ignore.case = T)]<-"Arc
hitecture"
h1b.data1$Imp_occupation[grep('information security',h1b.data1$SOC_NAME, ignore.case = T)]<-"Arc
hitecture"
h1b.data1$Imp_occupation[is.na(h1b.data1$Imp_occupation)] <- "others"
```

The above result covers almost 75% of the column and rest nan values are filled with others

```
h1b.data1<-separate(data = h1b.data1, col = WORKSITE, into = c("CITY", "STATE"), sep = ",")
```

As the H1b is more dependent on state rather than city therefore i divided the worksite into city and state as mentioned above.We will caputre state in our model

```
h1b.data1$CASE_STATUS<-ifelse(h1b.data1$CASE_STATUS %in% c("CERTIFIED"),"1","0")
```

I decided to use binary clssification for my model therefore converted my target variable into two classes of 0 and 1. 1 BEING CERTFIED AND 0 being denied

```
h1b.data1 <- h1b.data1 %>% select(-c (EMPLOYER_NAME, CITY, SOC_NAME, JOB_TITLE,YEAR,V1))
h1b.data1 <- h1b.data1 %>% select(-c (lon, lat))
```

Dropping column which are lot reuired in the datset for classification

```
h1b.data1$CASE_STATUS <- as.factor(h1b.data1$CASE_STATUS)
h1b.data1$FULL_TIME_POSITION <- as.factor(h1b.data1$FULL_TIME_POSITION)
h1b.data1$TOP_Sponsers <- as.factor(h1b.data1$TOP_Sponsers)
h1b.data1$STATE<- as.factor(h1b.data1$STATE)
h1b.data1$Imp_occupation <- as.factor(h1b.data1$Imp_occupation)
```

Converting the columns into factor for classification

```
summary(h1b.data1)
```

```
##  CASE_STATUS FULL_TIME_POSITION PREVAILING_WAGE            STATE
##  0: 2760     N: 3050            Min.   :        0    CALIFORNIA:5199
##  1:17240     Y:16950            1st Qu.:    80683    NEW YORK  :3012
##                                 Median :   106974    TEXAS     :1777
##                                 Mean   :   154051    ILLINOIS  : 997
##                                 3rd Qu.:   138133    FLORIDA   : 985
##                                 Max.   :329139200    WASHINGTON: 876
##                                                     (Other)    :7154
##      TOP_Sponsers      Imp_occupation
##   Top 5    :  313   Business  :    1
##  others    :19258   Finance   : 1742
##  Top 5     :   45   Management:  544
##  University:  384   others    :17713
##
##
##
```

As we see the summary we only have 6 variable left and now we will perform our Classiffication on them.

Machine learning model

```
inTrain <- createDataPartition(y = h1b.data1$CASE_STATUS,p = .7, list = FALSE,times=1 )
train1 <- h1b.data1[inTrain,]
test1<- h1b.data1[-inTrain,]
```

Spliting the dataet in to train and test 70% train and 30% test

SVM An svm is a discriminative clasisifer which uses hyperplane as its classifier. In other words, , the algorithm gives an optimal hyperplane which categorizes the data. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in both side.

1. SVM Linear - The following code we run 3 fold croos vaidation to get the value of c then we run Support vector linear.

```
train_control <- trainControl(method = 'cv', number = 3, verboseIter = T)
set.seed(100)
model1 <- train(CASE_STATUS~., data = train1, method = "svmLinear", trControl = train_control)
```

```
## + Fold1: C=1
## - Fold1: C=1
## + Fold2: C=1
```

```
## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.
```

```
## - Fold2: C=1
## + Fold3: C=1
## - Fold3: C=1
## Aggregating results
## Fitting final model on full training set
```

```
model1
```

```
## Support Vector Machines with Linear Kernel
##
## 14000 samples
##     5 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 9333, 9333, 9334
## Resampling results:
##
##   Accuracy   Kappa
##   0.8620715  0.00238693
##
## Tuning parameter 'C' was held constant at a value of 1
```

The cross validation provides the value of c=1

Now we make predictions based on the value we get from cross validation

BASED ON MY RESULTS FROM THE CROSS- VALIDATION, I RAN THE CONFUSION MATRIX AND WE CAN SEE THE RESULTS ARE REALLY GOOD. THIS GAVE ME A GOOD IDEA ON MY DATA SET AND WE CAN SEE THE ALGORITHM IS WORKING REALLY GREAT ON MY TEST DATA SET.

THE ACCURACY HERE IS ~ 86% AND WHICH IS REALLY GOOD AND THIS MODEL CAN BE USED FOR FUTURE PREDICTIONS. ALSO, THE MCNEMARS'S P-VALUE IS SIGNIFICANT AND WE CAN USE THIS AS OUR RESULT FOR FURTHER ANALYSIS.

SVM RAIDAL

```
model2 <- train(CASE_STATUS~., data = train1, method = "svmRadial", trControl = train_control)
```

```
## + Fold1: sigma=0.04058, C=0.25
## - Fold1: sigma=0.04058, C=0.25
## + Fold1: sigma=0.04058, C=0.50
## - Fold1: sigma=0.04058, C=0.50
## + Fold1: sigma=0.04058, C=1.00
## - Fold1: sigma=0.04058, C=1.00
## + Fold2: sigma=0.04058, C=0.25
## - Fold2: sigma=0.04058, C=0.25
## + Fold2: sigma=0.04058, C=0.50
## - Fold2: sigma=0.04058, C=0.50
## + Fold2: sigma=0.04058, C=1.00
## - Fold2: sigma=0.04058, C=1.00
## + Fold3: sigma=0.04058, C=0.25
## - Fold3: sigma=0.04058, C=0.25
## + Fold3: sigma=0.04058, C=0.50
## - Fold3: sigma=0.04058, C=0.50
## + Fold3: sigma=0.04058, C=1.00
## - Fold3: sigma=0.04058, C=1.00
## Aggregating results
## Selecting tuning parameters
## Fitting sigma = 0.0406, C = 0.25 on full training set
```

```
model2
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 14000 samples
##     5 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 9333, 9333, 9334
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa
##   0.25  0.8620000   0.0000000000
##   0.50  0.8615715  -0.0001079324
##   1.00  0.8611429   0.0005335296
##
## Tuning parameter 'sigma' was held constant at a value of 0.04058055
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 0.04058055 and C = 0.25.
```

```
model2test<- predict(object=model2,newdata=test1)
```

NOW, WHEN I TRIED TO RUN THE SAME PROBLEM WITH DIFFERENT DIFFERENT VALUE OF C, I CAN SEE THAT THE ACCURACY HAS INCREASED TO ~86%. hERE, AS PLAY MORE WITH POUR VALUES OF C, THE BETTER THE RESULTS WE SEE.

THIS GAVE MA GOOD IDEA ON THE TOPICS I COVERED IN THIS SUBJECT AND HOW I CAN USE THE SAME ALGORITHMS, IN THE REAL LIFE. IT IS REALLY A GOOD LEARNING

```
confusionMatrix(data=model2test,reference=test1$CASE_STATUS,positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0    0    0
##          1  828 5172
##
##                Accuracy : 0.862
##                  95% CI : (0.853, 0.8706)
##     No Information Rate : 0.862
##     P-Value [Acc > NIR] : 0.5093
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.000
##             Specificity : 0.000
##          Pos Pred Value : 0.862
##          Neg Pred Value :   NaN
##              Prevalence : 0.862
##          Detection Rate : 0.862
##    Detection Prevalence : 1.000
##       Balanced Accuracy : 0.500
##
##        'Positive' Class : 1
##
```

Random FOREST Random forest consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction

```
model3 <- train(CASE_STATUS~., data = train1, method = "rf", trControl = train_control)
```

```
## + Fold1: mtry= 2
## - Fold1: mtry= 2
## + Fold1: mtry=31
## - Fold1: mtry=31
## + Fold1: mtry=60
## - Fold1: mtry=60
## + Fold2: mtry= 2
## - Fold2: mtry= 2
## + Fold2: mtry=31
## - Fold2: mtry=31
## + Fold2: mtry=60
## - Fold2: mtry=60
## + Fold3: mtry= 2
## - Fold3: mtry= 2
## + Fold3: mtry=31
## - Fold3: mtry=31
## + Fold3: mtry=60
## - Fold3: mtry=60
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 2 on full training set
```

```
model3
```

```
## Random Forest
##
## 14000 samples
##     5 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 9334, 9333, 9333
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.8620000  0.00000000
##   31    0.8571430  0.02470343
##   60    0.8316429  0.11107099
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

```
model3test<- predict(object=model3,newdata=test1)
```

```
confusionMatrix(data=model3test,reference=test1$CASE_STATUS,positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0    0    0
##          1  828 5172
##
##                Accuracy : 0.862
##                  95% CI : (0.853, 0.8706)
##     No Information Rate : 0.862
##     P-Value [Acc > NIR] : 0.5093
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.000
##             Specificity : 0.000
##          Pos Pred Value : 0.862
##          Neg Pred Value :   NaN
##              Prevalence : 0.862
##          Detection Rate : 0.862
##    Detection Prevalence : 1.000
##       Balanced Accuracy : 0.500
##
##        'Positive' Class : 1
##
```

NOW, WHEN I TRIED TO RUN THE SAME PROBLEM With RANDOM FOREST, WE CAN SEE THAT THE ACCURACY COMES OUT TO BE 86.2 WHICH IS SIMILAR TO SVM LINEAR

FROM THE ABOVE MODEL THE BEST PREDICTION IS OF SVM RADIAL. WE CAN USE VARIOUS OTHER MODEL FOR FURTHER ANALYSIS SUCH AS XGBOOST, KNN etc

REFERENCES : https://towardsdatascience.com/how-much-do-data-scientists-make-cbd7ec2b458 (https://towardsdatascience.com/how-much-do-data-scientists-make-cbd7ec2b458)
https://towardsdatascience.com/predicting-h-1b-status-using-random-forest-dc199a6d254c (https://towardsdatascience.com/predicting-h-1b-status-using-random-forest-dc199a6d254c)
https://webpages.uncc.edu/sshinde5/ (https://webpages.uncc.edu/sshinde5/)