ML Model

Cohort B Team 3

3/21/2020

Load Library

```
library(dplyr)
library(ggplot2)
library(fastDummies)
library(caret)
library(MASS)
library(kernlab)
library(randomForest)
library(gbm)
```

Load the dataset

```
data <- read.csv("indeed_job_dataset.csv")
glimpse(data)
```

```
## Observations: 5,715
## Variables: 43
## $ X                           <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1...
## $ Job_Title                   <fct> "Data Scientist", "Data Scienti...
## $ Link                        <fct> https://www.indeed.com/rc/clk?j...
## $ Queried_Salary              <fct> <80000, <80000, <80000, <80000,...
## $ Job_Type                    <fct> data_scientist, data_scientist,...
## $ Skill                       <fct> "['SAP', 'SQL']", "['Machine Le...
## $ No_of_Skills                <int> 2, 5, 9, 1, 7, 6, 10, 3, 4, 6, ...
## $ Company                     <fct> Express Scripts, Money Mart Fin...
## $ No_of_Reviews               <dbl> 3301, NA, 62, 158, 495, 173, 30...
## $ No_of_Stars                 <dbl> 3.3, NA, 3.5, 4.3, 4.1, 4.3, 3....
## $ Date_Since_Posted           <int> 1, 15, 1, 30, 30, 30, 5, 10, 1,...
## $ Description                 <fct> "[<p><b>POSITION SUMMARY</b></p...
## $ Location                    <fct> MO, TX, OR, DC, TX, MD, NY, GA,...
## $ Company_Revenue             <fct> More than $10B (USD), , , , , ,...
## $ Company_Employees           <fct> "10,000+", "", "", "", "Less th...
## $ Company_Industry            <fct> Health Care, , , Government, Ba...
## $ python                      <int> 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1...
## $ sql                         <int> 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0...
## $ machine.learning            <int> 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1...
## $ r                           <int> 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1...
## $ hadoop                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ tableau                     <int> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0...
## $ sas                         <int> 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1...
## $ spark                       <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ java                          <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ Others                        <int> 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1...
## $ CA                            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ NY                            <int> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0...
## $ VA                            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ TX                            <int> 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0...
## $ MA                            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ IL                            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ WA                            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ MD                            <int> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0...
## $ DC                            <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0...
## $ NC                            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ Other_states                  <int> 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1...
## $ Consulting.and.Business.Services <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ Internet.and.Software         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ Banks.and.Financial.Services  <int> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0...
## $ Health.Care                   <int> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ Insurance                     <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ Other_industries              <int> 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1...
```

Create a new working data called my data by removing some columns

```
mydata <- data %>% dplyr::select(-X:-Link, -Skill, -Company, -Date_Since_Posted:-Location, -Company_Indu
dim(mydata)
```

```
## [1] 5715   34
```

EDA

```
head(mydata)
```

```
##   Queried_Salary       Job_Type No_of_Skills No_of_Reviews No_of_Stars
## 1         <80000 data_scientist            2          3301         3.3
## 2         <80000 data_scientist            5            NA          NA
## 3         <80000 data_scientist            9            62         3.5
## 4         <80000 data_scientist            1           158         4.3
## 5         <80000 data_scientist            7           495         4.1
## 6         <80000 data_scientist            6           173         4.3
##        Company_Revenue Company_Employees python sql machine.learning r
## 1 More than $10B (USD)           10,000+      0   1                0 0
## 2                                             1   1                1 1
## 3                                             1   1                0 1
## 4                                             0   0                0 0
## 5                      Less than 10,000      0   0                0 1
## 6                                             0   0                1 0
##   hadoop tableau sas spark java Others CA NY VA TX MA IL WA MD DC NC
## 1      0       0   0     0    0      1  0  0  0  0  0  0  0  0  0  0
## 2      0       0   1     0    0      0  0  0  0  1  0  0  0  0  0  0
## 3      0       0   1     0    0      1  0  0  0  0  0  0  0  0  0  0
## 4      0       0   0     0    0      1  0  0  0  0  0  0  0  0  1  0
## 5      0       1   0     0    0      1  0  0  0  1  0  0  0  0  0  0
## 6      0       0   0     0    0      1  0  0  0  0  0  0  0  1  0  0
##   Other_states Consulting.and.Business.Services Internet.and.Software
## 1            1                                0                     0
## 2            0                                0                     0
## 3            1                                0                     0
## 4            0                                0                     0
## 5            0                                0                     0
## 6            0                                0                     0
##   Banks.and.Financial.Services Health.Care Insurance Other_industries
## 1                            0           1         0                0
## 2                            0           0         0                0
## 3                            0           0         0                0
## 4                            0           0         0                1
## 5                            1           0         0                0
## 6                            0           0         0                0
```

```r
summary(mydata)
```

```
##       Queried_Salary           Job_Type     No_of_Skills
##  <80000       : 788   data_analyst  :1793   Min.   : 0.000
##  >160000      : 415   data_engineer :1379   1st Qu.: 4.000
##  100000-119999:1394   data_scientist:2543   Median : 7.000
##  120000-139999:1292                         Mean   : 7.804
##  140000-159999: 873                         3rd Qu.:11.000
##  80000-99999  : 953                         Max.   :20.000
##
##  No_of_Reviews     No_of_Stars               Company_Revenue
##  Min.   :     2   Min.   :1.300                         :3698
##  1st Qu.:    33   1st Qu.:3.700   $1B to $5B (USD)     : 314
##  Median :   387   Median :3.900   $5B to $10B (USD)    : 396
##  Mean   :  4311   Mean   :3.846   Less than $1B (USD) : 262
##  3rd Qu.:  2581   3rd Qu.:4.100   More than $10B (USD):1045
##  Max.   :157475   Max.   :5.000
##  NA's   :962      NA's   :962
##        Company_Employees     python             sql
##                   :2516    Min.   :0.0000   Min.   :0.0000
##  10,000+          :2004    1st Qu.:0.0000   1st Qu.:0.0000
##  Less than 10,000:1195    Median :1.0000   Median :1.0000
##                            Mean   :0.5818   Mean   :0.5431
##                            3rd Qu.:1.0000   3rd Qu.:1.0000
##                            Max.   :1.0000   Max.   :1.0000
##
##  machine.learning       r              hadoop          tableau
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.4019   Mean   :0.3909   Mean   :0.2999   Mean   :0.2163
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
```

```
##
##      sas             spark            java          Others
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:1.0000
##  Median :0.0000   Median :0.0000   Median :0.000   Median :1.0000
##  Mean   :0.1647   Mean   :0.2679   Mean   :0.259   Mean   :0.9015
##  3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.000   Max.   :1.0000
##
##       CA               NY               VA                TX
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.0000   Median :0.0000   Median :0.00000   Median :0.00000
##  Mean   :0.2441   Mean   :0.1052   Mean   :0.05844   Mean   :0.05757
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00000   Max.   :1.00000
##
##       MA                IL                WA                MD
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.04742   Mean   :0.04199   Mean   :0.03885   Mean   :0.02957
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##
##       DC               NC            Other_states
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.000
##  Median :0.0000   Median :0.00000   Median :0.000
##  Mean   :0.0245   Mean   :0.02432   Mean   :0.284
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:1.000
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.000
##
##  Consulting.and.Business.Services Internet.and.Software
##  Min.   :0.0000                   Min.   :0.0000
##  1st Qu.:0.0000                   1st Qu.:0.0000
##  Median :0.0000                   Median :0.0000
##  Mean   :0.1283                   Mean   :0.1132
##  3rd Qu.:0.0000                   3rd Qu.:0.0000
##  Max.   :1.0000                   Max.   :1.0000
##
##  Banks.and.Financial.Services  Health.Care        Insurance
##  Min.   :0.00000             Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000             1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000             Median :0.00000   Median :0.00000
##  Mean   :0.08031             Mean   :0.05932   Mean   :0.03972
##  3rd Qu.:0.00000             3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000             Max.   :1.00000   Max.   :1.00000
##
##  Other_industries
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.2486
##  3rd Qu.:0.0000
##  Max.   :1.0000
##
```

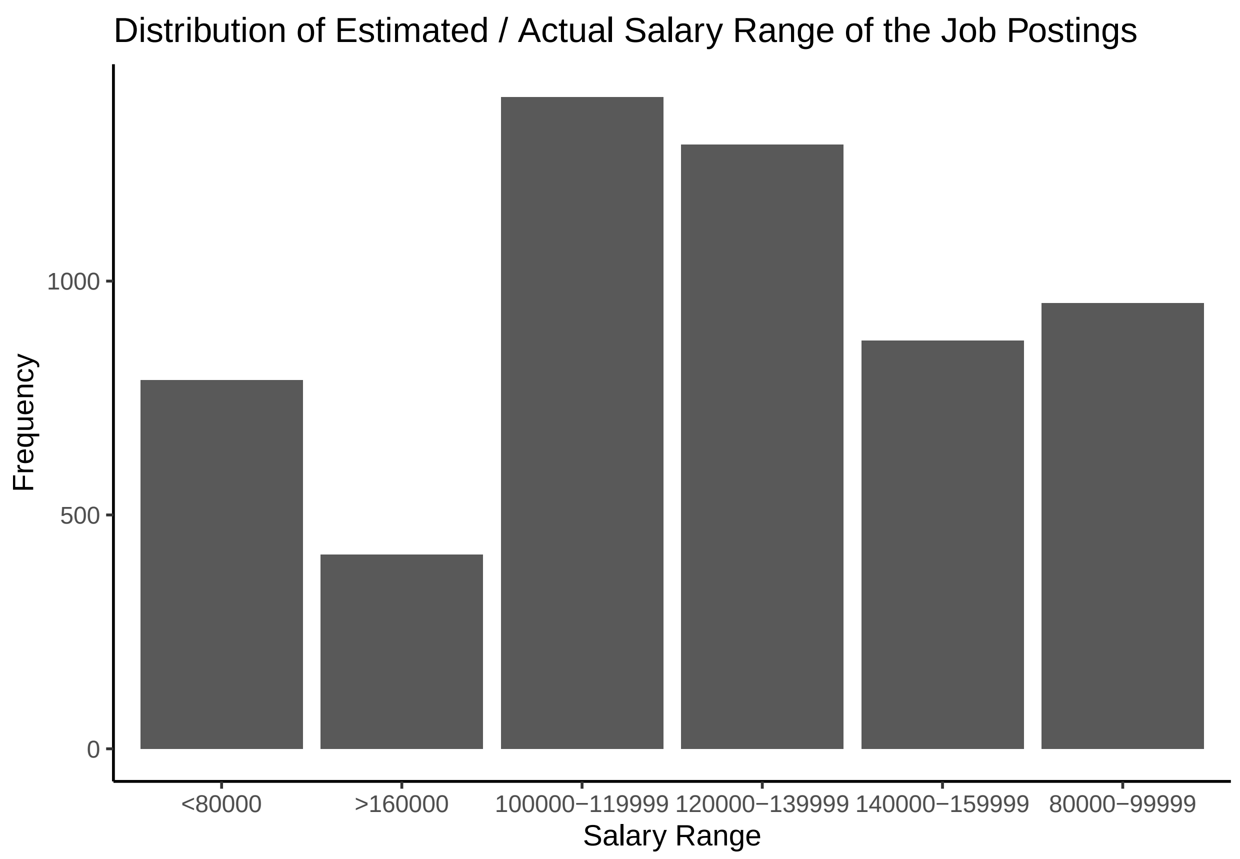
- 3 main job types: analyst, engineer, scientist
- No. of skills: Median - 7, Mean - 7.804, Range - 0 - 20
- 962 companies don't have any reviews/ ratings on Indeed
- Ineed does not have information on some companies revenue and number of employee information

Analysis on salary range

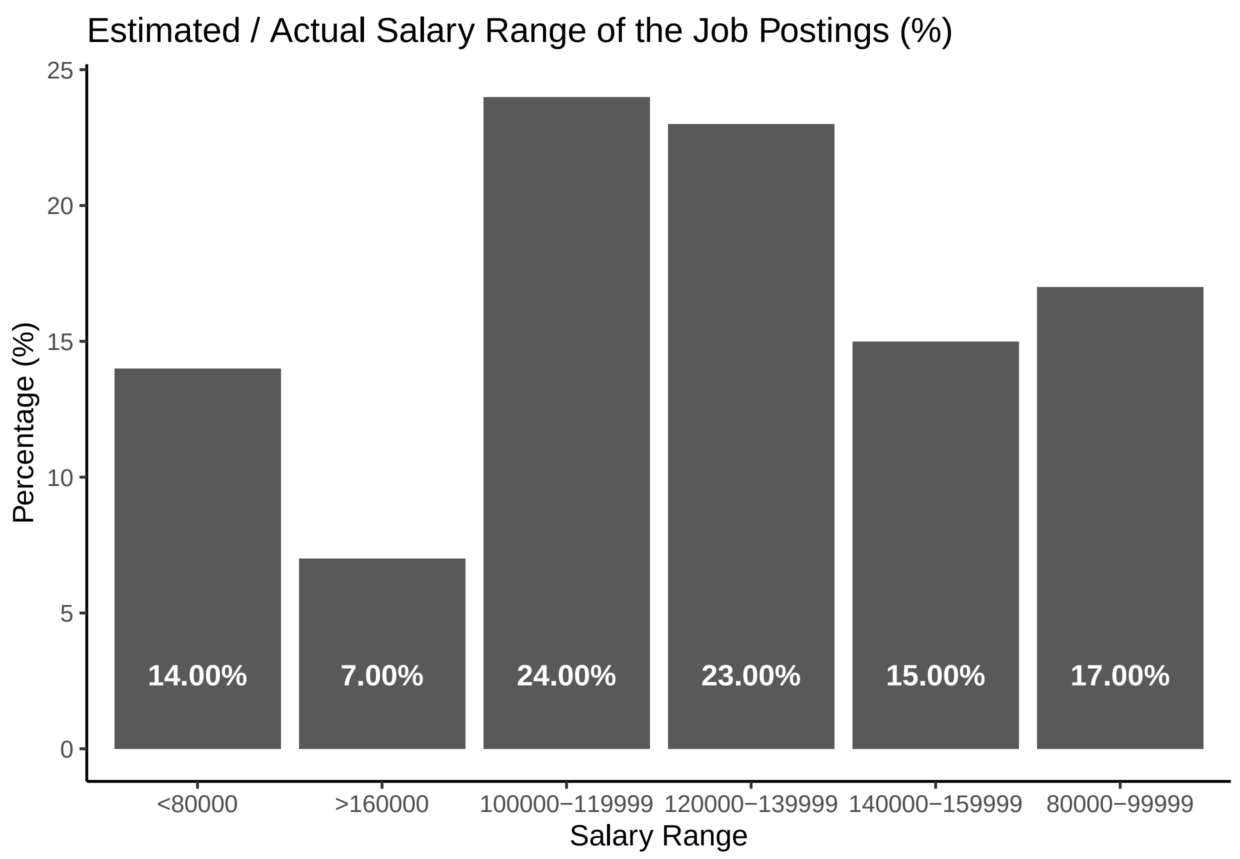
```
percentage <- prop.table(table(mydata$Queried_Salary)) * 100
cbind(freq=table(mydata$Queried_Salary), percentage=percentage)
```

```
##               freq percentage
## <80000         788  13.788276
## >160000        415   7.261592
## 100000-119999 1394  24.391951
## 120000-139999 1292  22.607174
## 140000-159999  873  15.275591
## 80000-99999    953  16.675416
```

```
# Count of each salary range
ggplot(mydata) +
  geom_histogram(aes(x = as.factor(Queried_Salary)),stat="count") +
  theme_classic() +
  labs(title = "Distribution of Estimated / Actual Salary Range of the Job Postings",
       x = "Salary Range", y = "Frequency")
```

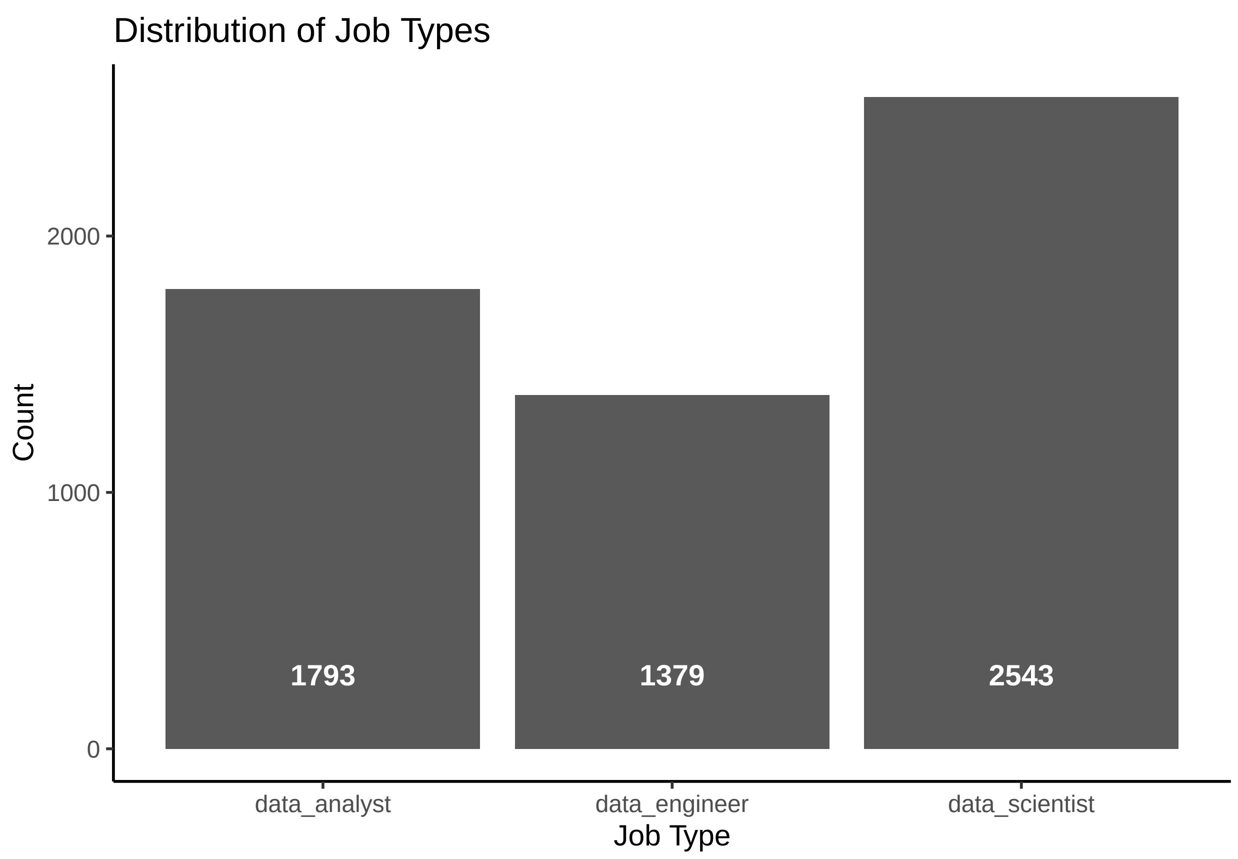


```r
# % of each salary range among the dataset
mydata %>%
  group_by(Queried_Salary) %>%
  summarize(count=n()) %>%
  mutate(perct = round(prop.table(count),2)*100) %>%
  ggplot(aes(x = Queried_Salary, y = perct)) +
  geom_histogram(stat = "identity")+
  geom_text(aes(x=Queried_Salary, y=0.01, label= sprintf("%.2f%%", perct)),
            hjust=0.5, vjust=-3, size=4,
            color="white", fontface = "bold") +
  theme_classic() +
  labs(x = "Salary Range", y="Percentage (%)",
       title = "Estimated / Actual Salary Range of the Job Postings (%)")
```



Other variables

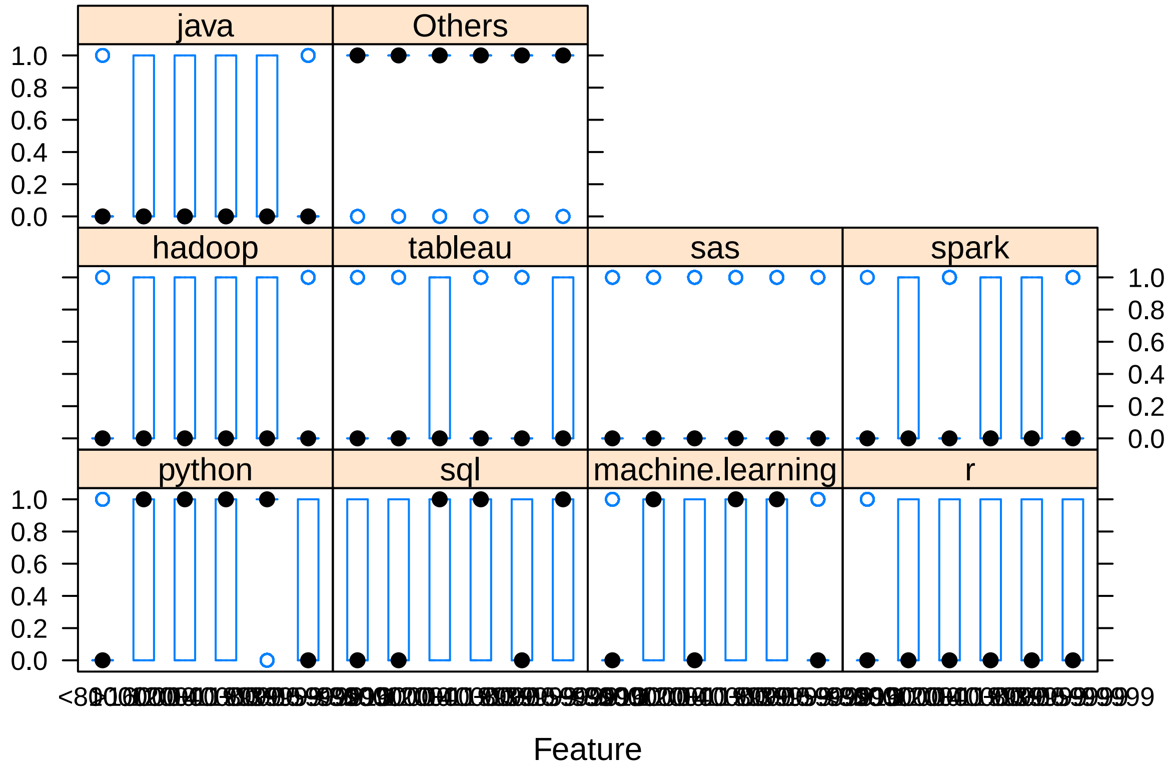
```r
# Count of each job type
mydata %>%
  group_by(Job_Type) %>%
  summarize(count = n()) %>%
  ggplot(aes(x = Job_Type, y = count)) +
  geom_bar(stat = "identity") +
  theme_classic() +
  geom_text(aes(x = Job_Type, y = 1, label = count),
            hjust = 0.5, vjust = -3, size = 4,
            color = "white", fontface = "bold") +
  labs(title = "Distribution of Job Types", x = "Job Type", y="Count")
```



```
## Multivariate Plots - look at the interactions between the variables
skills <- mydata[ ,8:17]
skills %>% head()
```

```
##   python sql machine.learning r hadoop tableau sas spark java Others
## 1      0   1                0 0      0       0   0     0    0      1
## 2      1   1                1 1      0       0   1     0    0      0
## 3      1   1                0 1      0       0   1     0    0      1
## 4      0   0                0 0      0       0   0     0    0      1
## 5      0   0                0 1      0       1   0     0    0      1
## 6      0   0                1 0      0       0   0     0    0      1
```

```
featurePlot(x=skills, y=mydata$Queried_Salary, plot="box")
```



Data Cleaning

```r
summary(mydata)
```

```
##        Queried_Salary           Job_Type     No_of_Skills
##  <80000        : 788   data_analyst  :1793   Min.   : 0.000
##  >160000       : 415   data_engineer :1379   1st Qu.: 4.000
##  100000-119999:1394   data_scientist:2543   Median : 7.000
##  120000-139999:1292                          Mean   : 7.804
##  140000-159999: 873                          3rd Qu.:11.000
##  80000-99999   : 953                          Max.   :20.000
##
##  No_of_Reviews     No_of_Stars                Company_Revenue
##  Min.   :      2   Min.   :1.300                          :3698
##  1st Qu.:     33   1st Qu.:3.700   $1B to $5B (USD)     : 314
##  Median :    387   Median :3.900   $5B to $10B (USD)    : 396
##  Mean   :   4311   Mean   :3.846   Less than $1B (USD) : 262
##  3rd Qu.:   2581   3rd Qu.:4.100   More than $10B (USD):1045
##  Max.   :157475   Max.   :5.000
##  NA's   :962      NA's   :962
##        Company_Employees      python            sql
##                  :2516    Min.   :0.0000    Min.   :0.0000
##  10,000+         :2004    1st Qu.:0.0000    1st Qu.:0.0000
##  Less than 10,000:1195    Median :1.0000   Median :1.0000
##                           Mean   :0.5818   Mean   :0.5431
##                           3rd Qu.:1.0000   3rd Qu.:1.0000
##                           Max.   :1.0000   Max.   :1.0000
##
##  machine.learning       r              hadoop          tableau
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.4019   Mean   :0.3909   Mean   :0.2999   Mean   :0.2163
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##
##       sas             spark             java          Others
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:1.0000
##  Median :0.0000   Median :0.0000   Median :0.000   Median :1.0000
##  Mean   :0.1647   Mean   :0.2679   Mean   :0.259   Mean   :0.9015
##  3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.000   Max.   :1.0000
##
##        CA               NY               VA                 TX
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.0000   Median :0.0000   Median :0.00000   Median :0.00000
##  Mean   :0.2441   Mean   :0.1052   Mean   :0.05844   Mean   :0.05757
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00000   Max.   :1.00000
##
##        MA                IL                WA                 MD
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.04742   Mean   :0.04199   Mean   :0.03885   Mean   :0.02957
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##
##        DC               NC            Other_states
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.000
##  Median :0.0000   Median :0.00000   Median :0.000
##  Mean   :0.0245   Mean   :0.02432   Mean   :0.284
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:1.000
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.000
##
##  Consulting.and.Business.Services Internet.and.Software
##  Min.   :0.0000                   Min.   :0.0000
##  1st Qu.:0.0000                   1st Qu.:0.0000
##  Median :0.0000                   Median :0.0000
##  Mean   :0.1283                   Mean   :0.1132
##  3rd Qu.:0.0000                   3rd Qu.:0.0000
##  Max.   :1.0000                   Max.   :1.0000
##
##  Banks.and.Financial.Services  Health.Care        Insurance
##  Min.   :0.00000              Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000              1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000              Median :0.00000   Median :0.00000
##  Mean   :0.08031              Mean   :0.05932   Mean   :0.03972
##  3rd Qu.:0.00000              3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000              Max.   :1.00000   Max.   :1.00000
##
##  Other_industries
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.2486
##  3rd Qu.:0.0000
##  Max.   :1.0000
##
```

```
# shows that Company_Revenue & Company_Employees have blank values

# fill those blank value with NA
# Company_Revenue
mydata$Company_Revenue <- as.character(mydata$Company_Revenue)
mydata$Company_Revenue[mydata$Company_Revenue == ""] <- "NA"
mydata$Company_Revenue <- as.factor(mydata$Company_Revenue)
summary(mydata$Company_Revenue)
```

```
##     $1B to $5B (USD)    $5B to $10B (USD)  Less than $1B (USD)
##                  314                  396                  262
## More than $10B (USD)                   NA
##                 1045                 3698
```

```
mydata$Company_Employees <- as.character(mydata$Company_Employees)
mydata$Company_Employees[mydata$Company_Employees == ""] <- "NA"
mydata$Company_Employees <- as.factor(mydata$Company_Employees)
summary(mydata$Company_Employees)
```

```
##         10,000+ Less than 10,000               NA
##            2004             1195             2516
```

```
# replace NAs with o for No_of_Reviews & No_of_Stars
mydata[is.na(mydata)] <- 0

# Check if there's any missing value in this dataset
sapply(mydata, function(x) sum(is.na(x)))
```

```
##                   Queried_Salary                         Job_Type
##                                0                                0
##                     No_of_Skills                    No_of_Reviews
##                                0                                0
##                      No_of_Stars                  Company_Revenue
##                                0                                0
##                Company_Employees                           python
##                                0                                0
##                              sql                 machine.learning
##                                0                                0
##                                r                           hadoop
##                                0                                0
##                          tableau                              sas
##                                0                                0
##                            spark                             java
##                                0                                0
##                           Others                               CA
##                                0                                0
##                               NY                               VA
##                                0                                0
##                               TX                               MA
##                                0                                0
##                               IL                               WA
##                                0                                0
##                               MD                               DC
##                                0                                0
##                               NC                     Other_states
##                                0                                0
## Consulting.and.Business.Services            Internet.and.Software
##                                0                                0
##     Banks.and.Financial.Services                      Health.Care
##                                0                                0
##                        Insurance                 Other_industries
##                                0                                0
```

Dummify the following columns

```
str(mydata) # check if the columns needed to be dumified are in factor forms
```

```
## 'data.frame':    5715 obs. of  34 variables:
##  $ Queried_Salary                  : Factor w/ 6 levels "<80000",">160000",..: 1 1 1 1 1 1 1 1 1 1 .
##  $ Job_Type                        : Factor w/ 3 levels "data_analyst",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ No_of_Skills                    : int  2 5 9 1 7 6 10 3 4 6 ...
##  $ No_of_Reviews                   : num  3301 0 62 158 495 ...
##  $ No_of_Stars                     : num  3.3 0 3.5 4.3 4.1 ...
##  $ Company_Revenue                 : Factor w/ 5 levels "$1B to $5B (USD)",..: 4 5 5 5 5 5 5 5 5 5 .
##  $ Company_Employees               : Factor w/ 3 levels "10,000+","Less than 10,000",..: 1 3 3 3 2 3
##  $ python                          : int  0 1 1 0 0 0 1 0 1 1 ...
##  $ sql                             : int  1 1 1 0 0 0 1 1 0 0 ...
##  $ machine.learning                : int  0 1 0 0 0 1 1 1 0 0 ...
##  $ r                               : int  0 1 1 0 1 0 1 1 1 1 ...
##  $ hadoop                          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ tableau                         : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ sas                             : int  0 1 1 0 0 0 0 0 0 0 ...
##  $ spark                           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ java                            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Others                          : int  1 0 1 1 1 1 1 0 1 1 ...
##  $ CA                              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ NY                              : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ VA                              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ TX                              : int  0 1 0 0 1 0 0 0 0 0 ...
##  $ MA                             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ IL                             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WA                             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MD                             : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ DC                             : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ NC                             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Other_states                   : int  1 0 1 0 0 0 0 1 1 1 ...
##  $ Consulting.and.Business.Services: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Internet.and.Software          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Banks.and.Financial.Services    : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Health.Care                    : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Insurance                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Other_industries               : int  0 0 0 1 0 0 1 0 1 1 ...
```

```r
mydata <-dummy_cols(mydata)

mydata <- mydata %>% dplyr::select(-Job_Type, -Company_Revenue, - Company_Revenue, - Company_Employees,
                            -"Queried_Salary_<80000": -"Queried_Salary_80000-99999" )
```

Change colnames

```r
colnames(mydata)
```

```
##  [1] "Queried_Salary"
##  [2] "No_of_Skills"
##  [3] "No_of_Reviews"
##  [4] "No_of_Stars"
##  [5] "python"
##  [6] "sql"
##  [7] "machine.learning"
##  [8] "r"
##  [9] "hadoop"
## [10] "tableau"
## [11] "sas"
## [12] "spark"
## [13] "java"
## [14] "Others"
## [15] "CA"
## [16] "NY"
## [17] "VA"
## [18] "TX"
## [19] "MA"
## [20] "IL"
## [21] "WA"
## [22] "MD"
## [23] "DC"
## [24] "NC"
## [25] "Other_states"
## [26] "Consulting.and.Business.Services"
## [27] "Internet.and.Software"
## [28] "Banks.and.Financial.Services"
## [29] "Health.Care"
## [30] "Insurance"
## [31] "Other_industries"
## [32] "Job_Type_data_analyst"
## [33] "Job_Type_data_engineer"
## [34] "Job_Type_data_scientist"
## [35] "Company_Revenue_$1B to $5B (USD)"
## [36] "Company_Revenue_$5B to $10B (USD)"
## [37] "Company_Revenue_Less than $1B (USD)"
## [38] "Company_Revenue_More than $10B (USD)"
## [39] "Company_Revenue_NA"
## [40] "Company_Employees_10,000+"
## [41] "Company_Employees_Less than 10,000"
## [42] "Company_Employees_NA"
```

```
mydata <- mydata %>% rename_all(tolower)

colnames(mydata)[colnames(mydata) == "queried_salary"] <- "salary"
colnames(mydata)[colnames(mydata) == "others"] <- "other_skills"

colnames(mydata)[colnames(mydata) == "ca"] <- "california"
colnames(mydata)[colnames(mydata) == "ny"] <- "new_york"
colnames(mydata)[colnames(mydata) == "va"] <- "virginia"
colnames(mydata)[colnames(mydata) == "tx"] <- "texas"
colnames(mydata)[colnames(mydata) == "ma"] <- "massachusetts"
colnames(mydata)[colnames(mydata) == "il"] <- "illinois"
colnames(mydata)[colnames(mydata) == "wa"] <- "washington"
colnames(mydata)[colnames(mydata) == "md"] <- "maryland"
colnames(mydata)[colnames(mydata) == "dc"] <- "dc"
colnames(mydata)[colnames(mydata) == "nc"] <- "north_carolina"

colnames(mydata)[colnames(mydata) == "job_type_data_analyst"] <- "data_analyst"
colnames(mydata)[colnames(mydata) == "job_type_data_engineer"] <- "data_engineer"
colnames(mydata)[colnames(mydata) == "job_type_data_scientist"] <- "data_scientist"


colnames(mydata)[colnames(mydata) == "company_revenue_$1b to $5b (usd)"] <- "revenue_$1bto$5b"
colnames(mydata)[colnames(mydata) == "company_revenue_$5b to $10b (usd)"] <- "revenue_$5bto$10b"
colnames(mydata)[colnames(mydata) == "company_revenue_less than $1b (usd)"] <- "revenue<$1b"
colnames(mydata)[colnames(mydata) == "company_revenue_more than $10b (usd)"] <- "revenue>$10b"
colnames(mydata)[colnames(mydata) == "company_revenue_na"] <- "revenue_na"

colnames(mydata)[colnames(mydata) == "company_employees_10,000+"] <- "employees>10k"
colnames(mydata)[colnames(mydata) == "company_employees_less than 10,000"] <- "employees<10k"
colnames(mydata)[colnames(mydata) == "company_employees_na"] <- "employees_na"
colnames(mydata)
```

```
##  [1] "salary"                            "no_of_skills"
##  [3] "no_of_reviews"                     "no_of_stars"
##  [5] "python"                            "sql"
##  [7] "machine.learning"                  "r"
##  [9] "hadoop"                            "tableau"
## [11] "sas"                               "spark"
## [13] "java"                              "other_skills"
## [15] "california"                        "new_york"
## [17] "virginia"                          "texas"
## [19] "massachusetts"                    "illinois"
## [21] "washington"                       "maryland"
## [23] "dc"                               "north_carolina"
## [25] "other_states"                     "consulting.and.business.services"
## [27] "internet.and.software"            "banks.and.financial.services"
## [29] "health.care"                      "insurance"
## [31] "other_industries"                 "data_analyst"
## [33] "data_engineer"                    "data_scientist"
## [35] "revenue_$1bto$5b"                 "revenue_$5bto$10b"
## [37] "revenue<$1b"                      "revenue>$10b"
## [39] "revenue_na"                       "employees>10k"
## [41] "employees<10k"                    "employees_na"
```

Building machine learning models

Split into the training and testing datasets

```r
levels(mydata$salary)
```

```
## [1] "<80000"        ">160000"       "100000-119999" "120000-139999"
## [5] "140000-159999" "80000-99999"
```

```r
# Determine sample size
set.seed(123456)

# create a list of 80% of the rows in the original dataset we can use for training
validation_index <- createDataPartition(mydata$salary, p=0.80, list=FALSE)
# select 20% of the data for validation
mydata_test <- mydata[-validation_index, ]
# use the remaining 80% of data to training and testing the models
mydata_train <- mydata[validation_index, ]

dim(mydata)
```

```
## [1] 5715   42
```

```r
dim(mydata_train)
```

```
## [1] 4575   42
```

```r
dim(mydata_test)
```

```
## [1] 1140   42
```

- Run algorithms using 10-fold cross validation

```r
control <- trainControl(method="cv", number=10)
metric <- "Accuracy"
```

Using the metric of "Accuracy" to evaluate machine learning models. This is a ratio of the number of correctly predicted instances in divided by the total number of instances in the dataset multiplied by 100 to give a percentage (e.g. 95% accurate).

Fit models a) Linear Discriminant Analysis (LDA)

```
fit.lda <- train(salary~., data=mydata_train, method="lda",
                  metric=metric, trControl=control)
```

b) StepwiseRegression

```
# library(MASS)
fit.stepwise <- train(salary~., data=mydata_train, method="stepLDA",
                       metric=metric, trControl=control)
```

```
## correctness rate: 0.34005;  in: "data_analyst";  variables (1): data_analyst
##
##  hr.elapsed min.elapsed sec.elapsed
##       0.000       0.000      15.259
##
## correctness rate: 0.33989;  in: "data_analyst";  variables (1): data_analyst
##
##  hr.elapsed min.elapsed sec.elapsed
##       0.000       0.000      14.944
##
## correctness rate: 0.33843;  in: "data_analyst";  variables (1): data_analyst
##
##  hr.elapsed min.elapsed sec.elapsed
##       0.000       0.000      16.032
##
## correctness rate: 0.33803;  in: "data_analyst";  variables (1): data_analyst
##
##  hr.elapsed min.elapsed sec.elapsed
##       0.000       0.000      16.206
##
## correctness rate: 0.33746;  in: "data_analyst";  variables (1): data_analyst
##
##  hr.elapsed min.elapsed sec.elapsed
##       0.000       0.000      14.864
##
## correctness rate: 0.33746;  in: "data_analyst";  variables (1): data_analyst
##
##  hr.elapsed min.elapsed sec.elapsed
##       0.000       0.000      14.975
##
## correctness rate: 0.33754;  in: "data_analyst";  variables (1): data_analyst
##
##  hr.elapsed min.elapsed sec.elapsed
##       0.000       0.000      13.859
##
## correctness rate: 0.33762;  in: "data_analyst";  variables (1): data_analyst
##
##  hr.elapsed min.elapsed sec.elapsed
##       0.000       0.000      15.084
##
## correctness rate: 0.33795;  in: "data_analyst";  variables (1): data_analyst
##
##  hr.elapsed min.elapsed sec.elapsed
##       0.000       0.000      12.839
##
## correctness rate: 0.33698;  in: "data_analyst";  variables (1): data_analyst
##
##  hr.elapsed min.elapsed sec.elapsed
##       0.000       0.000      14.109
##
## correctness rate: 0.33814;  in: "data_analyst";  variables (1): data_analyst
##
##  hr.elapsed min.elapsed sec.elapsed
##        0.00        0.00       15.73
```

c) k-Nearest Neighbors (kNN)

```
fit.knn <- train(salary~., data=mydata_train, method="knn",
                 metric=metric, trControl=control)
```

d) Support Vector Machines (SVM) with a linear kernel

```
# library(kernlab)
# takes a long time!
fit.svm <- train(salary~., data=mydata_train, method="svmRadial",
                 metric=metric, trControl=control)
```

e) Random Forest (RF)

```
# library(randomForest)
fit.rf <- train(salary~., data=mydata_train, method="rf",
                metric=metric, trControl=control)
```

f) boosted trees

```
# library(gbm)
fit.gbm <- train(salary~., data=mydata_train, method="gbm",
                 metric=metric, trControl=control)
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1056
##      2        1.7290             nan     0.1000    0.0826
##      3        1.6808             nan     0.1000    0.0645
##      4        1.6430             nan     0.1000    0.0520
##      5        1.6118             nan     0.1000    0.0404
##      6        1.5876             nan     0.1000    0.0339
##      7        1.5683             nan     0.1000    0.0255
##      8        1.5518             nan     0.1000    0.0247
##      9        1.5368             nan     0.1000    0.0231
##     10        1.5231             nan     0.1000    0.0170
##     20        1.4514             nan     0.1000    0.0045
##     40        1.3915             nan     0.1000   -0.0003
##     60        1.3590             nan     0.1000   -0.0005
##     80        1.3400             nan     0.1000   -0.0005
##    100        1.3257             nan     0.1000   -0.0011
##    120        1.3154             nan     0.1000   -0.0017
##    140        1.3069             nan     0.1000   -0.0009
##    150        1.3028             nan     0.1000   -0.0009
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1310
##      2        1.7173             nan     0.1000    0.1003
##      3        1.6594             nan     0.1000    0.0780
##      4        1.6151             nan     0.1000    0.0576
##      5        1.5801             nan     0.1000    0.0485
##      6        1.5508             nan     0.1000    0.0395
##      7        1.5269             nan     0.1000    0.0334
##      8        1.5064             nan     0.1000    0.0251
##      9        1.4892             nan     0.1000    0.0209
##     10        1.4740             nan     0.1000    0.0134
##     20        1.3934             nan     0.1000    0.0033
##     40        1.3256             nan     0.1000    0.0007
##     60        1.2893             nan     0.1000   -0.0016
##     80        1.2599             nan     0.1000   -0.0020
##    100        1.2375             nan     0.1000   -0.0019
##    120        1.2177             nan     0.1000   -0.0028
##    140        1.2040             nan     0.1000   -0.0040
##    150        1.1971             nan     0.1000   -0.0017
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1419
##      2        1.7091             nan     0.1000    0.1049
##      3        1.6494             nan     0.1000    0.0811
##      4        1.6016             nan     0.1000    0.0567
##      5        1.5647             nan     0.1000    0.0511
##      6        1.5318             nan     0.1000    0.0433
##      7        1.5047             nan     0.1000    0.0350
##      8        1.4821             nan     0.1000    0.0257
##      9        1.4649             nan     0.1000    0.0229
##     10        1.4501             nan     0.1000    0.0184
##     20        1.3566             nan     0.1000    0.0060
##     40        1.2763             nan     0.1000   -0.0003
##     60        1.2333             nan     0.1000   -0.0026
##     80        1.1963             nan     0.1000   -0.0016
##    100        1.1677             nan     0.1000   -0.0030
##    120        1.1426             nan     0.1000   -0.0030
##    140        1.1194             nan     0.1000   -0.0037
##    150        1.1085             nan     0.1000   -0.0033
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1059
##      2        1.7301             nan     0.1000    0.0818
##      3        1.6828             nan     0.1000    0.0674
##      4        1.6427             nan     0.1000    0.0513
##      5        1.6131             nan     0.1000    0.0450
##      6        1.5869             nan     0.1000    0.0356
##      7        1.5667             nan     0.1000    0.0295
##      8        1.5497             nan     0.1000    0.0241
##      9        1.5353             nan     0.1000    0.0209
##     10        1.5214             nan     0.1000    0.0190
##     20        1.4483             nan     0.1000    0.0072
##     40        1.3871             nan     0.1000    0.0017
##     60        1.3558             nan     0.1000   -0.0004
##     80        1.3372             nan     0.1000   -0.0012
##    100        1.3240             nan     0.1000   -0.0014
##    120        1.3142             nan     0.1000   -0.0012
##    140        1.3056             nan     0.1000   -0.0017
##    150        1.3023             nan     0.1000   -0.0010
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1342
##      2        1.7151             nan     0.1000    0.1013
##      3        1.6573             nan     0.1000    0.0720
##      4        1.6138             nan     0.1000    0.0593
##      5        1.5792             nan     0.1000    0.0489
##      6        1.5478             nan     0.1000    0.0388
##      7        1.5244             nan     0.1000    0.0318
##      8        1.5037             nan     0.1000    0.0296
##      9        1.4846             nan     0.1000    0.0199
##     10        1.4704             nan     0.1000    0.0161
##     20        1.3855             nan     0.1000    0.0026
##     40        1.3196             nan     0.1000   -0.0003
##     60        1.2843             nan     0.1000   -0.0008
##     80        1.2581             nan     0.1000   -0.0015
##    100        1.2386             nan     0.1000   -0.0020
##    120        1.2187             nan     0.1000   -0.0021
##    140        1.2035             nan     0.1000   -0.0030
##    150        1.1966             nan     0.1000   -0.0032
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1451
##      2        1.7051             nan     0.1000    0.1014
##      3        1.6464             nan     0.1000    0.0817
##      4        1.5991             nan     0.1000    0.0605
##      5        1.5614             nan     0.1000    0.0528
##      6        1.5279             nan     0.1000    0.0423
##      7        1.5011             nan     0.1000    0.0343
##      8        1.4789             nan     0.1000    0.0307
##      9        1.4600             nan     0.1000    0.0230
##     10        1.4445             nan     0.1000    0.0177
##     20        1.3554             nan     0.1000    0.0043
##     40        1.2793             nan     0.1000   -0.0015
##     60        1.2332             nan     0.1000   -0.0024
##     80        1.1999             nan     0.1000   -0.0023
##    100        1.1700             nan     0.1000   -0.0024
##    120        1.1450             nan     0.1000   -0.0028
##    140        1.1222             nan     0.1000   -0.0034
##      150         1.1122             nan     0.1000   -0.0032
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##        1         1.7918             nan     0.1000    0.1086
##        2         1.7311             nan     0.1000    0.0833
##        3         1.6830             nan     0.1000    0.0657
##        4         1.6450             nan     0.1000    0.0531
##        5         1.6128             nan     0.1000    0.0437
##        6         1.5884             nan     0.1000    0.0351
##        7         1.5673             nan     0.1000    0.0290
##        8         1.5493             nan     0.1000    0.0228
##        9         1.5346             nan     0.1000    0.0193
##       10         1.5232             nan     0.1000    0.0167
##       20         1.4497             nan     0.1000    0.0070
##       40         1.3909             nan     0.1000   -0.0002
##       60         1.3616             nan     0.1000   -0.0002
##       80         1.3431             nan     0.1000   -0.0015
##      100         1.3301             nan     0.1000   -0.0015
##      120         1.3203             nan     0.1000   -0.0011
##      140         1.3130             nan     0.1000   -0.0015
##      150         1.3093             nan     0.1000   -0.0013
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##        1         1.7918             nan     0.1000    0.1319
##        2         1.7128             nan     0.1000    0.0982
##        3         1.6571             nan     0.1000    0.0744
##        4         1.6151             nan     0.1000    0.0544
##        5         1.5821             nan     0.1000    0.0469
##        6         1.5540             nan     0.1000    0.0410
##        7         1.5292             nan     0.1000    0.0342
##        8         1.5087             nan     0.1000    0.0270
##        9         1.4912             nan     0.1000    0.0182
##       10         1.4765             nan     0.1000    0.0187
##       20         1.3965             nan     0.1000    0.0050
##       40         1.3293             nan     0.1000   -0.0008
##       60         1.2909             nan     0.1000   -0.0004
##       80         1.2632             nan     0.1000   -0.0022
##      100         1.2425             nan     0.1000   -0.0025
##      120         1.2256             nan     0.1000   -0.0014
##      140         1.2095             nan     0.1000   -0.0022
##      150         1.2026             nan     0.1000   -0.0029
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##        1         1.7918             nan     0.1000    0.1399
##        2         1.7091             nan     0.1000    0.1001
##        3         1.6491             nan     0.1000    0.0780
##        4         1.6004             nan     0.1000    0.0623
##        5         1.5622             nan     0.1000    0.0474
##        6         1.5309             nan     0.1000    0.0387
##        7         1.5059             nan     0.1000    0.0357
##        8         1.4831             nan     0.1000    0.0267
##        9         1.4649             nan     0.1000    0.0266
##       10         1.4475             nan     0.1000    0.0158
##       20         1.3631             nan     0.1000    0.0053
##     40        1.2852             nan     0.1000   -0.0005
##     60        1.2387             nan     0.1000   -0.0028
##     80        1.2064             nan     0.1000   -0.0026
##    100        1.1773             nan     0.1000   -0.0034
##    120        1.1534             nan     0.1000   -0.0033
##    140        1.1316             nan     0.1000   -0.0025
##    150        1.1224             nan     0.1000   -0.0033
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1050
##      2        1.7305             nan     0.1000    0.0804
##      3        1.6834             nan     0.1000    0.0651
##      4        1.6465             nan     0.1000    0.0506
##      5        1.6169             nan     0.1000    0.0410
##      6        1.5936             nan     0.1000    0.0346
##      7        1.5727             nan     0.1000    0.0283
##      8        1.5558             nan     0.1000    0.0202
##      9        1.5419             nan     0.1000    0.0214
##     10        1.5278             nan     0.1000    0.0176
##     20        1.4552             nan     0.1000    0.0038
##     40        1.3975             nan     0.1000    0.0016
##     60        1.3687             nan     0.1000   -0.0008
##     80        1.3509             nan     0.1000   -0.0010
##    100        1.3380             nan     0.1000   -0.0009
##    120        1.3277             nan     0.1000   -0.0010
##    140        1.3198             nan     0.1000   -0.0013
##    150        1.3166             nan     0.1000   -0.0012
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1254
##      2        1.7181             nan     0.1000    0.0974
##      3        1.6629             nan     0.1000    0.0702
##      4        1.6215             nan     0.1000    0.0658
##      5        1.5835             nan     0.1000    0.0443
##      6        1.5572             nan     0.1000    0.0377
##      7        1.5338             nan     0.1000    0.0304
##      8        1.5131             nan     0.1000    0.0243
##      9        1.4970             nan     0.1000    0.0246
##     10        1.4807             nan     0.1000    0.0158
##     20        1.3994             nan     0.1000    0.0034
##     40        1.3355             nan     0.1000   -0.0006
##     60        1.2991             nan     0.1000   -0.0022
##     80        1.2739             nan     0.1000   -0.0014
##    100        1.2527             nan     0.1000   -0.0020
##    120        1.2358             nan     0.1000   -0.0030
##    140        1.2201             nan     0.1000   -0.0035
##    150        1.2126             nan     0.1000   -0.0025
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1371
##      2        1.7097             nan     0.1000    0.1012
##      3        1.6489             nan     0.1000    0.0747
##      4        1.6013             nan     0.1000    0.0552
##      5        1.5662             nan     0.1000    0.0454
##      6        1.5369             nan     0.1000    0.0422
##      7        1.5100             nan     0.1000    0.0349
##      8        1.4877             nan     0.1000    0.0281
##      9        1.4696             nan     0.1000    0.0194
##     10        1.4555             nan     0.1000    0.0175
##     20        1.3670             nan     0.1000    0.0053
##     40        1.2914             nan     0.1000   -0.0004
##     60        1.2456             nan     0.1000   -0.0025
##     80        1.2117             nan     0.1000   -0.0022
##    100        1.1861             nan     0.1000   -0.0022
##    120        1.1628             nan     0.1000   -0.0029
##    140        1.1420             nan     0.1000   -0.0026
##    150        1.1334             nan     0.1000   -0.0033
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1071
##      2        1.7292             nan     0.1000    0.0824
##      3        1.6797             nan     0.1000    0.0651
##      4        1.6419             nan     0.1000    0.0531
##      5        1.6117             nan     0.1000    0.0439
##      6        1.5860             nan     0.1000    0.0336
##      7        1.5664             nan     0.1000    0.0288
##      8        1.5487             nan     0.1000    0.0243
##      9        1.5342             nan     0.1000    0.0196
##     10        1.5217             nan     0.1000    0.0178
##     20        1.4495             nan     0.1000    0.0053
##     40        1.3913             nan     0.1000    0.0003
##     60        1.3612             nan     0.1000   -0.0005
##     80        1.3428             nan     0.1000   -0.0014
##    100        1.3303             nan     0.1000   -0.0018
##    120        1.3204             nan     0.1000   -0.0023
##    140        1.3113             nan     0.1000   -0.0011
##    150        1.3079             nan     0.1000   -0.0017
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1348
##      2        1.7133             nan     0.1000    0.0978
##      3        1.6561             nan     0.1000    0.0768
##      4        1.6114             nan     0.1000    0.0596
##      5        1.5757             nan     0.1000    0.0484
##      6        1.5480             nan     0.1000    0.0393
##      7        1.5243             nan     0.1000    0.0293
##      8        1.5044             nan     0.1000    0.0257
##      9        1.4881             nan     0.1000    0.0198
##     10        1.4735             nan     0.1000    0.0174
##     20        1.3926             nan     0.1000    0.0047
##     40        1.3257             nan     0.1000   -0.0011
##     60        1.2891             nan     0.1000   -0.0010
##     80        1.2628             nan     0.1000   -0.0012
##    100        1.2410             nan     0.1000   -0.0023
##    120        1.2235             nan     0.1000   -0.0016
##    140        1.2076             nan     0.1000   -0.0022
##    150        1.2008             nan     0.1000   -0.0027
##
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1405
##      2        1.7099             nan     0.1000    0.1034
##      3        1.6481             nan     0.1000    0.0803
##      4        1.5979             nan     0.1000    0.0588
##      5        1.5611             nan     0.1000    0.0517
##      6        1.5290             nan     0.1000    0.0339
##      7        1.5042             nan     0.1000    0.0307
##      8        1.4841             nan     0.1000    0.0281
##      9        1.4649             nan     0.1000    0.0230
##     10        1.4484             nan     0.1000    0.0168
##     20        1.3603             nan     0.1000    0.0050
##     40        1.2835             nan     0.1000   -0.0002
##     60        1.2362             nan     0.1000   -0.0005
##     80        1.2013             nan     0.1000   -0.0026
##    100        1.1715             nan     0.1000   -0.0021
##    120        1.1455             nan     0.1000   -0.0030
##    140        1.1246             nan     0.1000   -0.0026
##    150        1.1132             nan     0.1000   -0.0031
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1068
##      2        1.7288             nan     0.1000    0.0810
##      3        1.6811             nan     0.1000    0.0627
##      4        1.6450             nan     0.1000    0.0544
##      5        1.6157             nan     0.1000    0.0419
##      6        1.5906             nan     0.1000    0.0347
##      7        1.5705             nan     0.1000    0.0289
##      8        1.5532             nan     0.1000    0.0252
##      9        1.5377             nan     0.1000    0.0212
##     10        1.5247             nan     0.1000    0.0152
##     20        1.4490             nan     0.1000    0.0054
##     40        1.3922             nan     0.1000    0.0016
##     60        1.3610             nan     0.1000    0.0000
##     80        1.3431             nan     0.1000   -0.0008
##    100        1.3296             nan     0.1000   -0.0002
##    120        1.3191             nan     0.1000   -0.0014
##    140        1.3110             nan     0.1000   -0.0012
##    150        1.3074             nan     0.1000   -0.0011
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1320
##      2        1.7174             nan     0.1000    0.0957
##      3        1.6603             nan     0.1000    0.0757
##      4        1.6143             nan     0.1000    0.0555
##      5        1.5819             nan     0.1000    0.0512
##      6        1.5516             nan     0.1000    0.0401
##      7        1.5272             nan     0.1000    0.0332
##      8        1.5065             nan     0.1000    0.0243
##      9        1.4900             nan     0.1000    0.0220
##     10        1.4758             nan     0.1000    0.0204
##     20        1.3947             nan     0.1000    0.0051
##     40        1.3298             nan     0.1000    0.0008
##     60        1.2906             nan     0.1000   -0.0012
##     80        1.2649             nan     0.1000   -0.0028
##    100        1.2439             nan     0.1000   -0.0011
##    120        1.2270             nan     0.1000   -0.0019
##    140        1.2114             nan     0.1000   -0.0032
##    150        1.2042             nan     0.1000   -0.0028
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1382
##      2        1.7087             nan     0.1000    0.1011
##      3        1.6468             nan     0.1000    0.0798
##      4        1.5983             nan     0.1000    0.0610
##      5        1.5592             nan     0.1000    0.0478
##      6        1.5282             nan     0.1000    0.0363
##      7        1.5042             nan     0.1000    0.0334
##      8        1.4825             nan     0.1000    0.0263
##      9        1.4642             nan     0.1000    0.0213
##     10        1.4491             nan     0.1000    0.0186
##     20        1.3607             nan     0.1000    0.0033
##     40        1.2863             nan     0.1000    0.0007
##     60        1.2394             nan     0.1000   -0.0015
##     80        1.2054             nan     0.1000    0.0005
##    100        1.1759             nan     0.1000   -0.0032
##    120        1.1525             nan     0.1000   -0.0028
##    140        1.1342             nan     0.1000   -0.0027
##    150        1.1236             nan     0.1000   -0.0034
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1075
##      2        1.7309             nan     0.1000    0.0818
##      3        1.6842             nan     0.1000    0.0674
##      4        1.6468             nan     0.1000    0.0541
##      5        1.6175             nan     0.1000    0.0427
##      6        1.5924             nan     0.1000    0.0370
##      7        1.5711             nan     0.1000    0.0260
##      8        1.5548             nan     0.1000    0.0244
##      9        1.5392             nan     0.1000    0.0190
##     10        1.5262             nan     0.1000    0.0160
##     20        1.4517             nan     0.1000    0.0071
##     40        1.3926             nan     0.1000    0.0007
##     60        1.3622             nan     0.1000    0.0001
##     80        1.3436             nan     0.1000   -0.0017
##    100        1.3307             nan     0.1000   -0.0012
##    120        1.3214             nan     0.1000   -0.0010
##    140        1.3131             nan     0.1000   -0.0010
##    150        1.3098             nan     0.1000   -0.0022
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1343
##      2        1.7163             nan     0.1000    0.0955
##      3        1.6596             nan     0.1000    0.0768
##      4        1.6157             nan     0.1000    0.0572
##      5        1.5806             nan     0.1000    0.0462
##      6        1.5522             nan     0.1000    0.0364
##      7        1.5294             nan     0.1000    0.0307
##      8        1.5087             nan     0.1000    0.0276
##      9        1.4918             nan     0.1000    0.0235
##     10        1.4765             nan     0.1000    0.0212
##     20        1.3925             nan     0.1000    0.0050
##     40        1.3252             nan     0.1000    0.0002
##     60        1.2902             nan     0.1000   -0.0014
##     80        1.2655             nan     0.1000   -0.0002
##    100        1.2444             nan     0.1000   -0.0021
##    120        1.2263             nan     0.1000   -0.0022
##    140        1.2109             nan     0.1000   -0.0019
##    150        1.2032             nan     0.1000   -0.0020
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1422
##      2        1.7102             nan     0.1000    0.1043
##      3        1.6492             nan     0.1000    0.0820
##      4        1.6002             nan     0.1000    0.0559
##      5        1.5642             nan     0.1000    0.0517
##      6        1.5333             nan     0.1000    0.0423
##      7        1.5065             nan     0.1000    0.0311
##      8        1.4840             nan     0.1000    0.0279
##      9        1.4661             nan     0.1000    0.0235
##     10        1.4488             nan     0.1000    0.0160
##     20        1.3597             nan     0.1000    0.0048
##     40        1.2833             nan     0.1000    0.0002
##     60        1.2355             nan     0.1000   -0.0015
##     80        1.2003             nan     0.1000   -0.0018
##    100        1.1729             nan     0.1000   -0.0028
##    120        1.1486             nan     0.1000   -0.0043
##    140        1.1287             nan     0.1000   -0.0011
##    150        1.1190             nan     0.1000   -0.0025
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1036
##      2        1.7318             nan     0.1000    0.0808
##      3        1.6845             nan     0.1000    0.0643
##      4        1.6486             nan     0.1000    0.0525
##      5        1.6174             nan     0.1000    0.0420
##      6        1.5933             nan     0.1000    0.0317
##      7        1.5733             nan     0.1000    0.0311
##      8        1.5552             nan     0.1000    0.0220
##      9        1.5408             nan     0.1000    0.0211
##     10        1.5279             nan     0.1000    0.0177
##     20        1.4561             nan     0.1000    0.0052
##     40        1.3942             nan     0.1000   -0.0003
##     60        1.3621             nan     0.1000   -0.0006
##     80        1.3425             nan     0.1000   -0.0006
##    100        1.3293             nan     0.1000   -0.0008
##    120        1.3194             nan     0.1000   -0.0008
##    140        1.3116             nan     0.1000   -0.0008
##    150        1.3079             nan     0.1000   -0.0012
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1252
##        2        1.7172             nan     0.1000    0.0960
##        3        1.6604             nan     0.1000    0.0702
##        4        1.6178             nan     0.1000    0.0553
##        5        1.5838             nan     0.1000    0.0463
##        6        1.5565             nan     0.1000    0.0400
##        7        1.5317             nan     0.1000    0.0336
##        8        1.5111             nan     0.1000    0.0250
##        9        1.4931             nan     0.1000    0.0208
##       10        1.4788             nan     0.1000    0.0198
##       20        1.3969             nan     0.1000    0.0050
##       40        1.3298             nan     0.1000   -0.0007
##       60        1.2937             nan     0.1000   -0.0007
##       80        1.2689             nan     0.1000   -0.0014
##      100        1.2477             nan     0.1000   -0.0019
##      120        1.2293             nan     0.1000   -0.0030
##      140        1.2129             nan     0.1000   -0.0011
##      150        1.2054             nan     0.1000   -0.0029
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##        1        1.7918             nan     0.1000    0.1376
##        2        1.7119             nan     0.1000    0.1040
##        3        1.6505             nan     0.1000    0.0703
##        4        1.6046             nan     0.1000    0.0629
##        5        1.5644             nan     0.1000    0.0495
##        6        1.5349             nan     0.1000    0.0390
##        7        1.5088             nan     0.1000    0.0314
##        8        1.4866             nan     0.1000    0.0296
##        9        1.4661             nan     0.1000    0.0221
##       10        1.4517             nan     0.1000    0.0186
##       20        1.3601             nan     0.1000    0.0028
##       40        1.2837             nan     0.1000   -0.0004
##       60        1.2395             nan     0.1000   -0.0028
##       80        1.2084             nan     0.1000   -0.0021
##      100        1.1809             nan     0.1000   -0.0020
##      120        1.1555             nan     0.1000   -0.0021
##      140        1.1337             nan     0.1000   -0.0029
##      150        1.1251             nan     0.1000   -0.0036
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##        1        1.7918             nan     0.1000    0.1075
##        2        1.7305             nan     0.1000    0.0848
##        3        1.6818             nan     0.1000    0.0654
##        4        1.6445             nan     0.1000    0.0546
##        5        1.6122             nan     0.1000    0.0420
##        6        1.5886             nan     0.1000    0.0347
##        7        1.5675             nan     0.1000    0.0292
##        8        1.5508             nan     0.1000    0.0256
##        9        1.5356             nan     0.1000    0.0209
##       10        1.5223             nan     0.1000    0.0155
##       20        1.4508             nan     0.1000    0.0061
##       40        1.3909             nan     0.1000    0.0007
##       60        1.3600             nan     0.1000    0.0004
##       80        1.3403             nan     0.1000   -0.0012
##      100        1.3251             nan     0.1000   -0.0008
##    120        1.3152             nan     0.1000   -0.0010
##    140        1.3072             nan     0.1000   -0.0014
##    150        1.3032             nan     0.1000   -0.0012
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1239
##      2        1.7179             nan     0.1000    0.1003
##      3        1.6593             nan     0.1000    0.0770
##      4        1.6137             nan     0.1000    0.0615
##      5        1.5780             nan     0.1000    0.0483
##      6        1.5482             nan     0.1000    0.0371
##      7        1.5249             nan     0.1000    0.0343
##      8        1.5040             nan     0.1000    0.0248
##      9        1.4878             nan     0.1000    0.0215
##     10        1.4728             nan     0.1000    0.0160
##     20        1.3898             nan     0.1000    0.0050
##     40        1.3233             nan     0.1000    0.0000
##     60        1.2866             nan     0.1000   -0.0006
##     80        1.2595             nan     0.1000   -0.0025
##    100        1.2388             nan     0.1000   -0.0038
##    120        1.2223             nan     0.1000   -0.0027
##    140        1.2079             nan     0.1000   -0.0020
##    150        1.2007             nan     0.1000   -0.0029
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1432
##      2        1.7090             nan     0.1000    0.1056
##      3        1.6453             nan     0.1000    0.0830
##      4        1.5963             nan     0.1000    0.0611
##      5        1.5578             nan     0.1000    0.0524
##      6        1.5249             nan     0.1000    0.0418
##      7        1.4985             nan     0.1000    0.0315
##      8        1.4777             nan     0.1000    0.0254
##      9        1.4596             nan     0.1000    0.0228
##     10        1.4434             nan     0.1000    0.0166
##     20        1.3518             nan     0.1000    0.0029
##     40        1.2796             nan     0.1000   -0.0006
##     60        1.2350             nan     0.1000   -0.0015
##     80        1.2000             nan     0.1000   -0.0008
##    100        1.1708             nan     0.1000   -0.0027
##    120        1.1467             nan     0.1000   -0.0018
##    140        1.1270             nan     0.1000   -0.0033
##    150        1.1183             nan     0.1000   -0.0030
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1080
##      2        1.7308             nan     0.1000    0.0826
##      3        1.6830             nan     0.1000    0.0659
##      4        1.6444             nan     0.1000    0.0531
##      5        1.6136             nan     0.1000    0.0407
##      6        1.5882             nan     0.1000    0.0357
##      7        1.5674             nan     0.1000    0.0286
##      8        1.5501             nan     0.1000    0.0199
##      9        1.5359             nan     0.1000    0.0197
##     10        1.5238             nan     0.1000    0.0181
##     20        1.4502             nan     0.1000    0.0040
##     40        1.3877             nan     0.1000    0.0027
##     60        1.3572             nan     0.1000   -0.0009
##     80        1.3392             nan     0.1000   -0.0008
##    100        1.3257             nan     0.1000   -0.0011
##    120        1.3152             nan     0.1000   -0.0007
##    140        1.3065             nan     0.1000   -0.0008
##    150        1.3031             nan     0.1000   -0.0013
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1344
##      2        1.7152             nan     0.1000    0.1002
##      3        1.6586             nan     0.1000    0.0746
##      4        1.6150             nan     0.1000    0.0638
##      5        1.5779             nan     0.1000    0.0484
##      6        1.5484             nan     0.1000    0.0370
##      7        1.5261             nan     0.1000    0.0325
##      8        1.5050             nan     0.1000    0.0270
##      9        1.4883             nan     0.1000    0.0181
##     10        1.4749             nan     0.1000    0.0202
##     20        1.3917             nan     0.1000    0.0032
##     40        1.3247             nan     0.1000    0.0003
##     60        1.2888             nan     0.1000   -0.0014
##     80        1.2593             nan     0.1000   -0.0013
##    100        1.2395             nan     0.1000   -0.0032
##    120        1.2238             nan     0.1000   -0.0025
##    140        1.2095             nan     0.1000   -0.0017
##    150        1.2032             nan     0.1000   -0.0024
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1400
##      2        1.7086             nan     0.1000    0.1069
##      3        1.6468             nan     0.1000    0.0781
##      4        1.5986             nan     0.1000    0.0596
##      5        1.5615             nan     0.1000    0.0498
##      6        1.5298             nan     0.1000    0.0393
##      7        1.5034             nan     0.1000    0.0340
##      8        1.4821             nan     0.1000    0.0265
##      9        1.4644             nan     0.1000    0.0273
##     10        1.4463             nan     0.1000    0.0156
##     20        1.3552             nan     0.1000    0.0042
##     40        1.2799             nan     0.1000    0.0001
##     60        1.2320             nan     0.1000   -0.0015
##     80        1.1981             nan     0.1000   -0.0021
##    100        1.1695             nan     0.1000   -0.0015
##    120        1.1457             nan     0.1000   -0.0030
##    140        1.1251             nan     0.1000   -0.0018
##    150        1.1149             nan     0.1000   -0.0044
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.7918             nan     0.1000    0.1402
##      2        1.7078             nan     0.1000    0.1024
##      3        1.6465             nan     0.1000    0.0802
##      4        1.5982             nan     0.1000    0.0647
##      5        1.5608             nan     0.1000    0.0484
##      6        1.5301             nan     0.1000    0.0394
##      7        1.5056             nan     0.1000    0.0331
##      8        1.4841             nan     0.1000    0.0274
##      9        1.4658             nan     0.1000    0.0214
##     10        1.4504             nan     0.1000    0.0185
##     20        1.3645             nan     0.1000    0.0053
##     40        1.2885             nan     0.1000   -0.0014
##     60        1.2447             nan     0.1000   -0.0013
##     80        1.2133             nan     0.1000   -0.0033
##    100        1.1856             nan     0.1000   -0.0020
##    120        1.1604             nan     0.1000   -0.0021
##    140        1.1399             nan     0.1000   -0.0024
##    150        1.1289             nan     0.1000   -0.0024
```

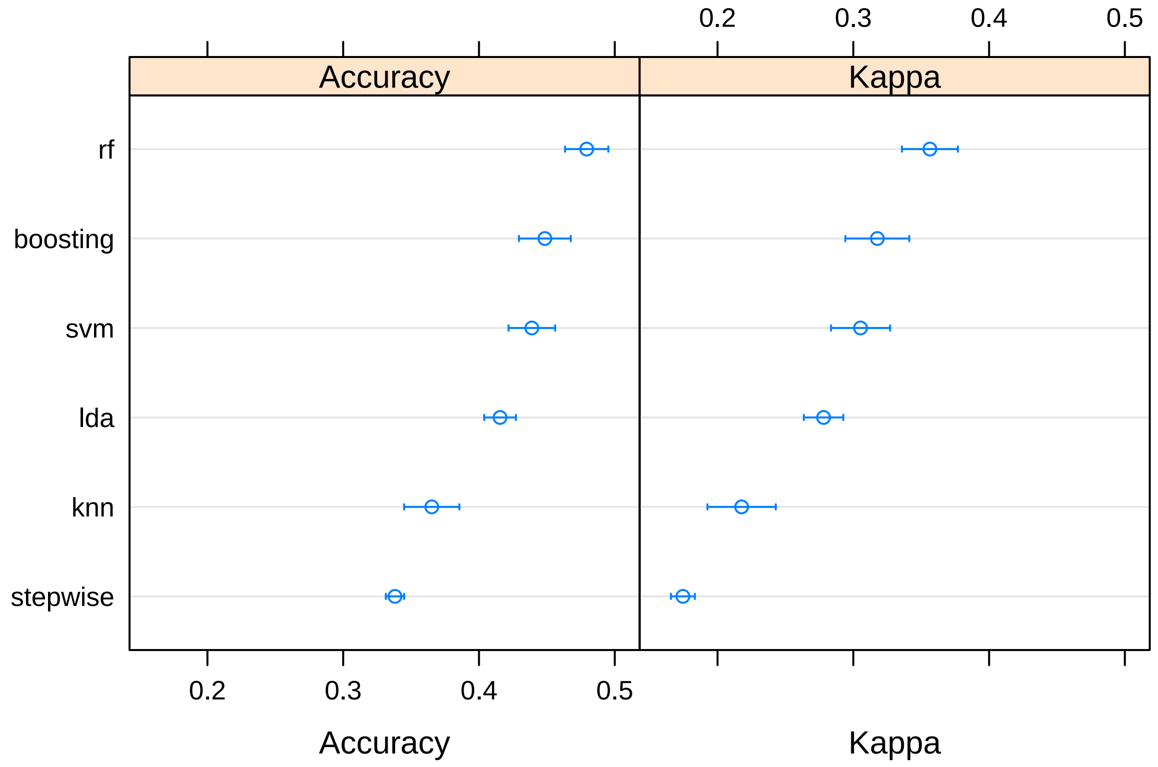
Summarize accuracy of models

```
results <- resamples(list(lda=fit.lda, stepwise = fit.stepwise, knn=fit.knn,
                          svm=fit.svm, rf=fit.rf, boosting=fit.gbm))
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: lda, stepwise, knn, svm, rf, boosting
## Number of resamples: 10
##
## Accuracy
##               Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lda      0.3820961 0.4057922 0.4185785 0.4155315 0.4259249 0.4385965    0
## stepwise 0.3209607 0.3364369 0.3413320 0.3381342 0.3440561 0.3485839    0
## knn      0.3129103 0.3615532 0.3719912 0.3652455 0.3847080 0.3951965    0
## svm      0.4008715 0.4231807 0.4338863 0.4389338 0.4556424 0.4792123    0
## rf       0.4529540 0.4609053 0.4731660 0.4793530 0.4989143 0.5152838    0
## boosting 0.4030501 0.4341156 0.4443231 0.4485436 0.4636788 0.5000000    0
##
## Kappa
##               Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lda      0.2360323 0.2672413 0.2807399 0.2781064 0.2919314 0.3079933    0
## stepwise 0.1517155 0.1727967 0.1785506 0.1744939 0.1822881 0.1879855    0
## knn      0.1544766 0.2113540 0.2263848 0.2177373 0.2410690 0.2580762    0
## svm      0.2552043 0.2860404 0.2984541 0.3053184 0.3264215 0.3555371    0
## rf       0.3215840 0.3305906 0.3509224 0.3564135 0.3795398 0.4031978    0
## boosting 0.2656429 0.2957417 0.3130207 0.3177086 0.3392095 0.3817013    0
```

Compare accuracy of models

```
dotplot(results)
```



Confidence Level: 0.95

As the grpah above shows, Random forest is the most arrucate.

Summary of the best model

```
print(fit.rf)
```

```
## Random Forest
##
## 4575 samples
##   41 predictor
##    6 classes: '<80000', '>160000', '100000-119999', '120000-139999', '140000-159999', '80000-99999'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4116, 4119, 4118, 4118, 4118, 4115, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.4572856  0.3234076
##   21    0.4793530  0.3564135
##   41    0.4747683  0.3505625
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 21.
```

Estimate the best model on testing dataset

```r
predictions <- predict(fit.rf, mydata_test)
confusionMatrix(predictions, mydata_test$salary)
```

```
## Confusion Matrix and Statistics
##
##                  Reference
## Prediction        <80000 >160000 100000-119999 120000-139999 140000-159999
##   <80000             105       3            25             2             0
##   >160000              3      25             5             6            13
##   100000-119999       11       7           145            74            24
##   120000-139999        3      15            48           121            61
##   140000-159999        2      30            19            45            74
##   80000-99999         33       3            36            10             2
##                  Reference
## Prediction        80000-99999
##   <80000                   43
##   >160000                   4
##   100000-119999            32
##   120000-139999            20
##   140000-159999             5
##   80000-99999              86
##
## Overall Statistics
##
##                Accuracy : 0.4877
##                  95% CI : (0.4583, 0.5172)
##     No Information Rate : 0.2439
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3681
##
##  Mcnemar's Test P-Value : 0.004122
##
## Statistics by Class:
##
##                        Class: <80000 Class: >160000 Class: 100000-119999
## Sensitivity                  0.66879        0.30120               0.5216
## Specificity                  0.92574        0.97067               0.8283
## Pos Pred Value               0.58989        0.44643               0.4949
## Neg Pred Value               0.94595        0.94649               0.8430
## Prevalence                   0.13772        0.07281               0.2439
## Detection Rate               0.09211        0.02193               0.1272
## Detection Prevalence         0.15614        0.04912               0.2570
## Balanced Accuracy            0.79726        0.63594               0.6749
##                        Class: 120000-139999 Class: 140000-159999
## Sensitivity                          0.4690              0.42529
## Specificity                          0.8333              0.89545
## Pos Pred Value                       0.4515              0.42286
## Neg Pred Value                       0.8429              0.89637
## Prevalence                           0.2263              0.15263
## Detection Rate                     0.1061             0.06491
## Detection Prevalence               0.2351             0.15351
## Balanced Accuracy                  0.6512             0.66037
##                      Class: 80000-99999
## Sensitivity                     0.45263
## Specificity                     0.91158
## Pos Pred Value                  0.50588
## Neg Pred Value                  0.89278
## Prevalence                      0.16667
## Detection Rate                  0.07544
## Detection Prevalence            0.14912
## Balanced Accuracy               0.68211
```

Create prediction based on MSBA students