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
```

```
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
##
```

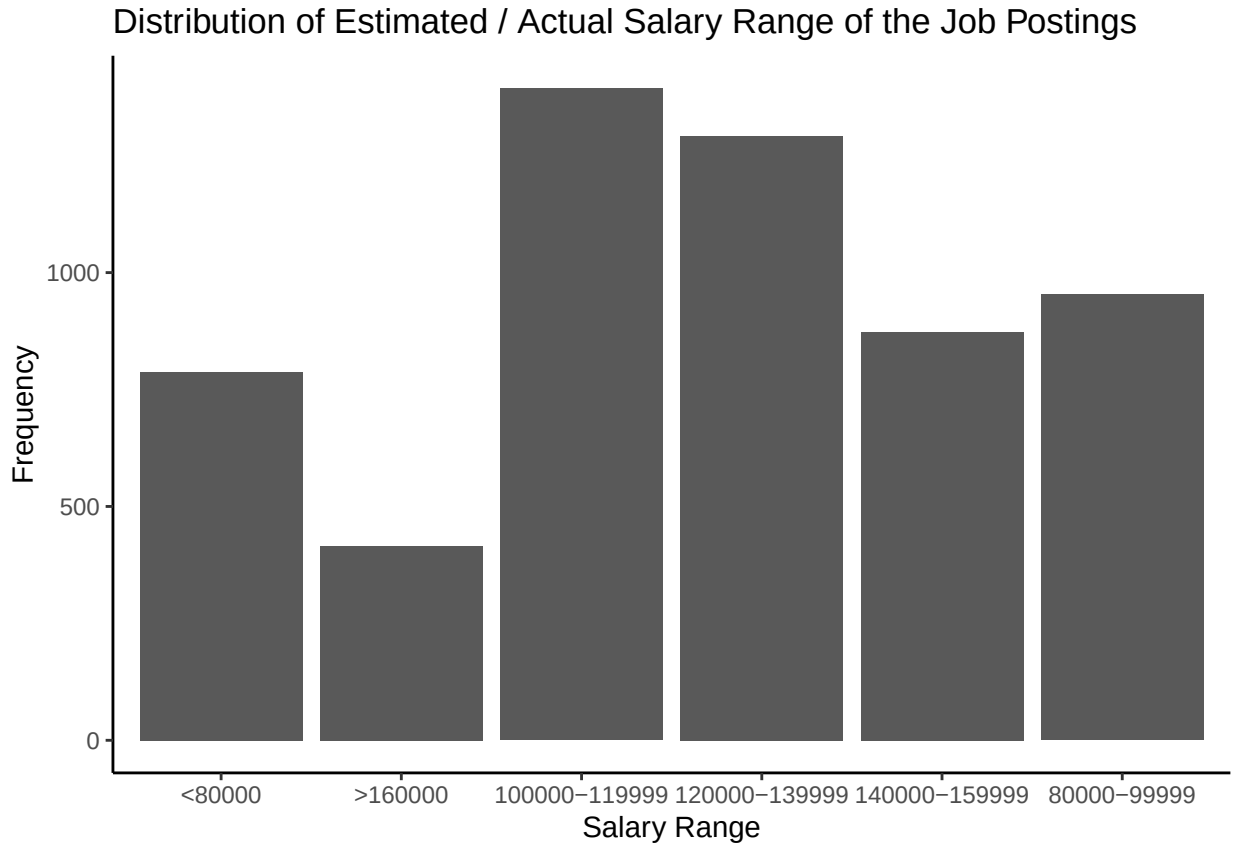
- 3 main job types: analyst, engineer, scientist
- No. of skills: Median - 7, Mean - 7.804, Range - 0 - 20
- 962 companies don't have any reviews/ ratings on Indeed
- Ineed does not have information on some companies revenue and number of employee information

Analysis on salary range

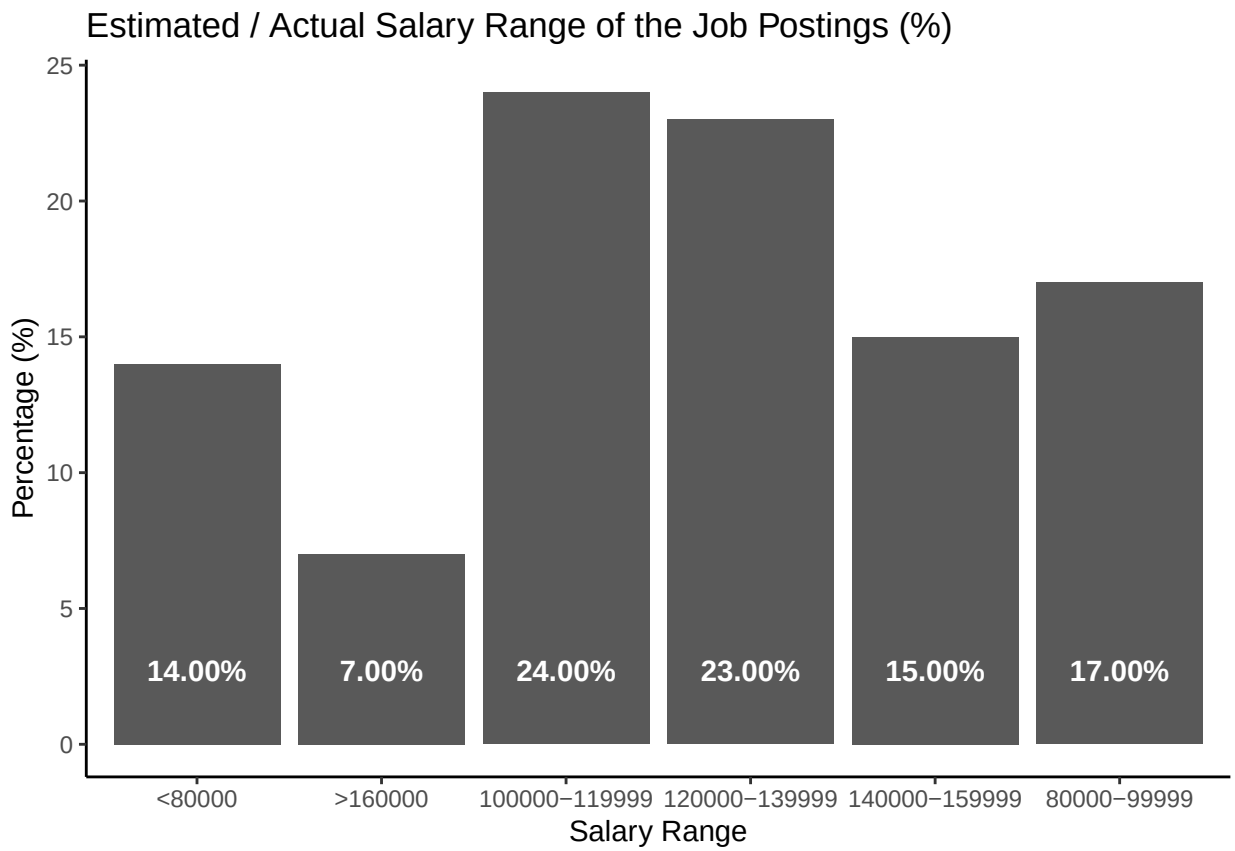
```
percentage <- prop.table(table(mydata$Queried_Salary)) * 100
cbind(freq=table(mydata$Queried_Salary), percentage=percentage)
```

```
##                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")
```

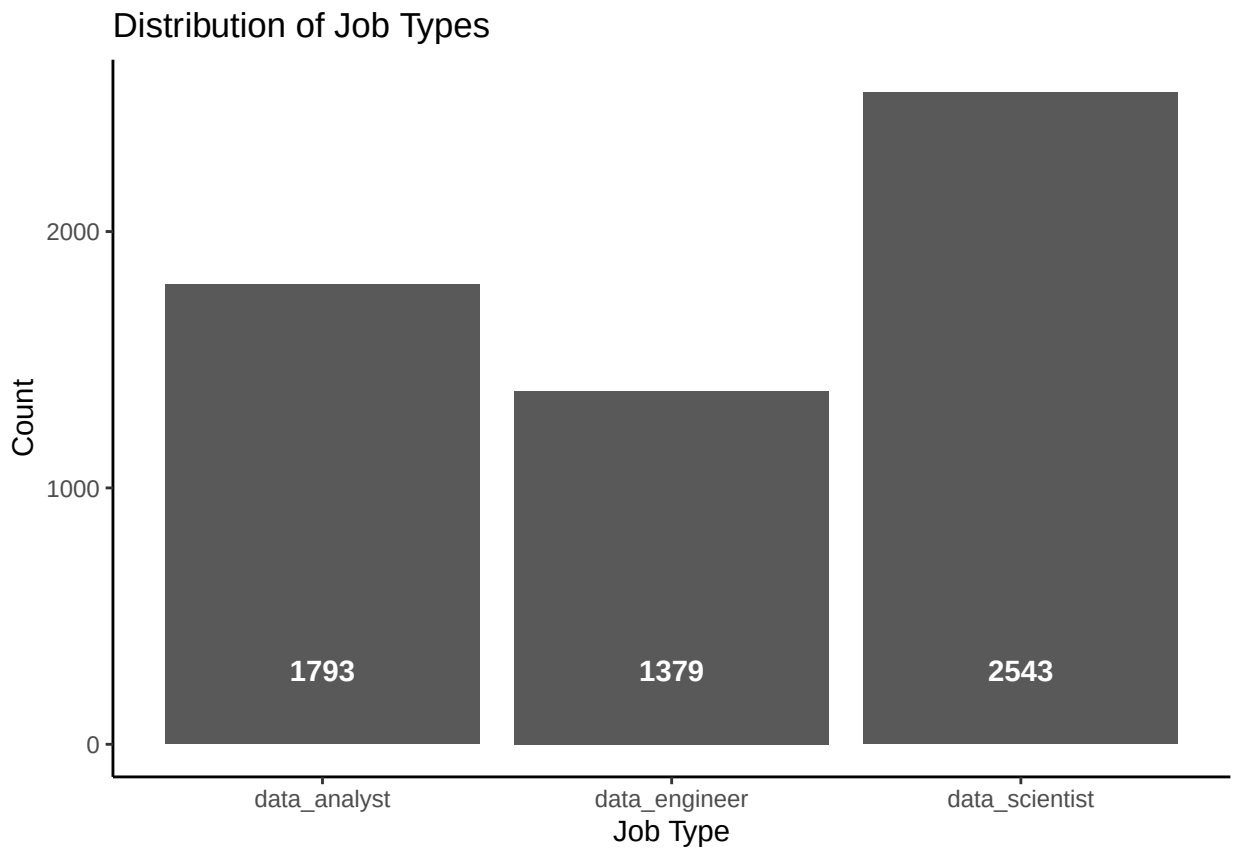



```
# % of each salary range among the dataset
mydata %>%
  group_by(Queried_Salary) %>%
  summarize(count=n()) %>%
  mutate(perct = round(prop.table(count),2)*100) %>%
  ggplot(aes(x = Queried_Salary, y = perct)) +
  geom_histogram(stat = "identity")+
  geom_text(aes(x=Queried_Salary, y=0.01, label= 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

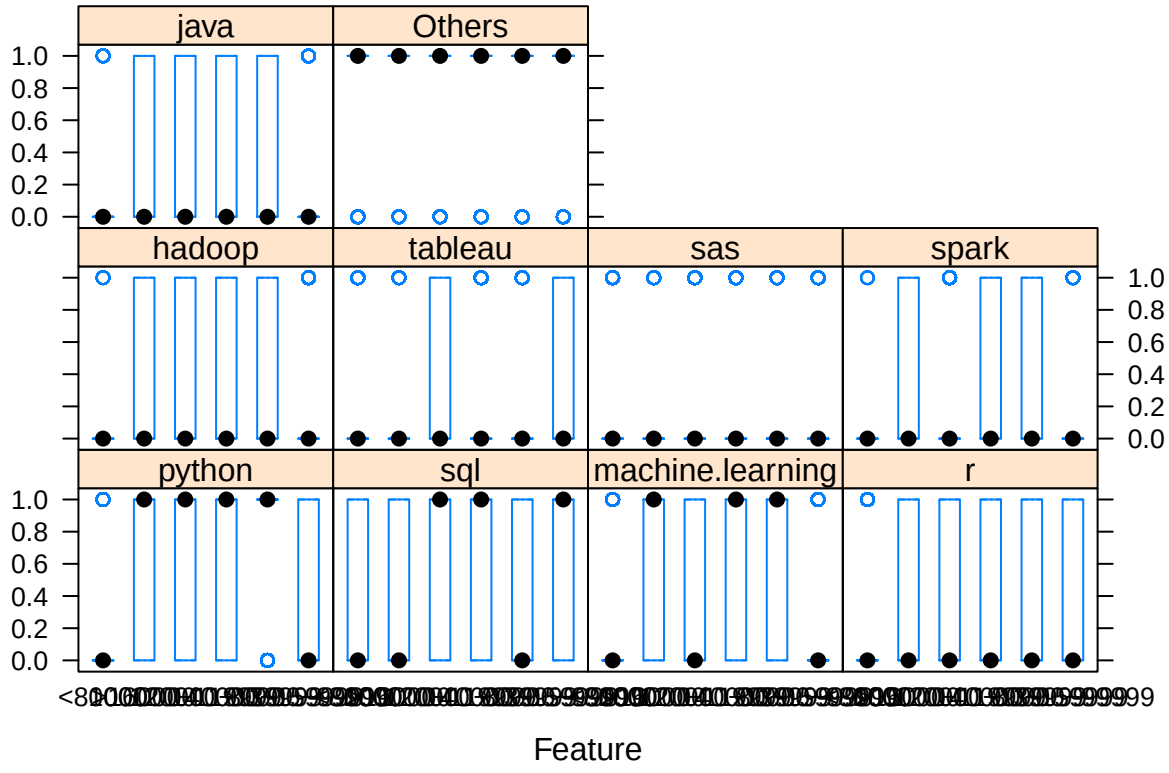
```
# Count of each job type
mydata %>%
  group_by(Job_Type) %>%
  summarize(count = n()) %>%
  ggplot(aes(x = Job_Type, y = count)) +
  geom_bar(stat = "identity") +
  theme_classic() +
  geom_text(aes(x = Job_Type, y = 1, label = count),
            hjust = 0.5, vjust = -3, size = 4,
            color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

```
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
##
```

```r
# 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
```

```r
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
```

```r
# 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 ...
```

```
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

```
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

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

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

```
# 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
```

```
dim(mydata_train)
```

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

```
dim(mydata_test)
```

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

- Run algorithms using 10-fold cross validation

```
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

c) k-Nearest Neighbors (kNN)

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

e) Random Forest (RF)

f) boosted trees

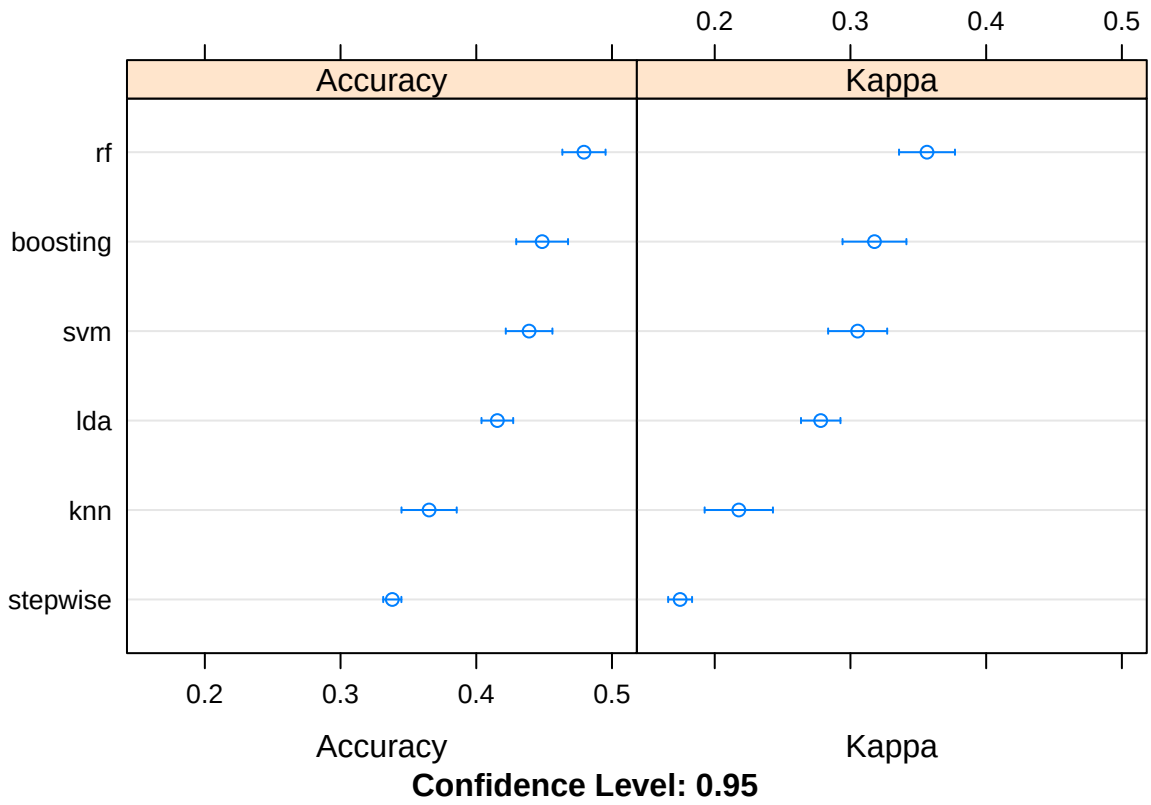
Summarize accuracy of models

```
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

```r
dotplot(results)
```



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

Summary of the best model

```r
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

```
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