HR Analytics Promotion Recommendation

Libraries Used

```r
library(data.table)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(caret)
```

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

```
## Loading required package: ggplot2
```

```r
library(ggplot2)
library(plyr)
```

```
## -------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## -------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```r
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(tree)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(MVA)
```

```
## Loading required package: HSAUR2
```

```
## Loading required package: tools
```

```r
library(htmltools)
library(base)
library(mlr)
```

```
## Loading required package: ParamHelpers
```

```
##
## Attaching package: 'mlr'
```

```
## The following object is masked from 'package:caret':
##
##     train
```

```r
library(FSelector)
library(ROSE)
```

```
## Loaded ROSE 0.0-3
```

```r
library(rpart)
library(regclass)
```

```
## Loading required package: bestglm
```

```
## Loading required package: leaps
```

```
## Loading required package: VGAM
```

```
## Loading required package: stats4
```

```
## Loading required package: splines
```

```
##
## Attaching package: 'VGAM'
```

```
## The following object is masked from 'package:caret':
##
##     predictors
```

```
## Important regclass change from 1.3:
## All functions that had a . in the name now have an _
## all.correlations -> all_correlations, cor.demo -> cor_demo, etc.
```

```
##
## Attaching package: 'regclass'
```

```
## The following object is masked from 'package:lattice':
##
##     qq
```

```r
library(e1071)
```

```
##
## Attaching package: 'e1071'
```

```
## The following object is masked from 'package:mlr':
##
##     impute
```

```r
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
library(DMwR)
```

```
## Loading required package: grid
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
##
## Attaching package: 'DMwR'
```

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

```r
library(randomForest)
library(ggplot2)
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, mutate, rename, summarise
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

Exploratory Data Analysis

Data Importing

```
setwd('C:/Users/harsh/Desktop/MITA/Fall 2019 Sem 2/DAV/Datasets/')
hr_analytics= read.csv("HR Analytics.csv", stringsAsFactors=FALSE, header=T, na.strings=c(""))
```

There are 14 attributes in the data set and 54808 observations

Categorical Variables

1. employee_id
2. department
3. region
4. education
5. gender
6. recruitment_channel
7. no_of_trainings
8. age
9. previous_year_rating 10.KPIs_met >80% 11.awards_won?

Quantitative Variables

1. length_of_service
2. avg_training_score

Target Variables

1. is_promoted

Converting dataframe into data table for flexibility

```
setDT(hr_analytics)
```

Checking for NA Values in the data set, column 9 which is previous_year_rating is having NA values

```
grep('NA',hr_analytics)
```

```
## [1] 4 9
```

Addressing NA’s

length of service where previous year rating is NA, seems like since person is joined recently previous year rating is not available

```
relation<-hr_analytics[is.na(previous_year_rating),.(length_of_service,previous_year_rating)]
unique(relation$length_of_service)
```

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

Replacing NA values in previous year rating with zeros

```
hr_analytics[is.na(previous_year_rating),previous_year_rating:=0]
```

```
unique(hr_analytics$previous_year_rating)
```

```
## [1] 5 3 1 4 0 2
```

```
str(hr_analytics)
```

```
## Classes 'data.table' and 'data.frame':   54808 obs. of  14 variables:
##  $ employee_id         : int  65438 65141 7513 2542 48945 58896 20379 16290 73202 28911 ...
##  $ department          : chr  "Sales & Marketing" "Operations" "Sales & Marketing" "Sales & Marketing" .
..
##  $ region              : chr  "region_7" "region_22" "region_19" "region_23" ...
##  $ education           : chr  "Master's & above" "Bachelor's" "Bachelor's" "Bachelor's" ...
##  $ gender              : chr  "f" "m" "m" "m" ...
##  $ recruitment_channel : chr  "sourcing" "other" "sourcing" "other" ...
##  $ no_of_trainings     : int  1 1 1 2 1 2 1 1 1 1 ...
##  $ age                 : int  35 30 34 39 45 31 31 33 28 32 ...
##  $ previous_year_rating: int  5 5 3 1 3 3 3 3 4 5 ...
##  $ length_of_service   : int  8 4 7 10 2 7 5 6 5 5 ...
##  $ KPIs_met..80.       : int  1 0 0 0 0 0 0 0 0 1 ...
##  $ awards_won.         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ avg_training_score  : int  49 60 50 50 73 85 59 63 83 54 ...
##  $ is_promoted         : int  0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Converting categorical columns into factors for better analysis

```
hr_analytics[,employee_id:=factor(employee_id)]
hr_analytics[,department:=factor(department)]
hr_analytics[,region:=factor(region)]
hr_analytics[,gender:=factor(gender,levels=c('m','f'),labels=c(0,1))]
hr_analytics[,recruitment_channel:=factor(recruitment_channel)]
hr_analytics[,KPIs_met..80.:=factor(KPIs_met..80.)]
hr_analytics[,awards_won.:=factor(awards_won.)]
hr_analytics[,previous_year_rating:=factor(previous_year_rating)]
```

```
str(hr_analytics$age)
```

```
##  int [1:54808] 35 30 34 39 45 31 31 33 28 32 ...
```

*** Converting education into factor and adding NA as a level for better analysis ***

```
hr_analytics$education<-addNA(hr_analytics$education)
```

```
levels(hr_analytics$education)
```

```
## [1] "Bachelor's"       "Below Secondary"  "Master's & above"
## [4] NA
```

EDA for categorical variables

Set Column Classes

```
factcols<-c(1:7,9,11,12,14)
numcols<-setdiff(1:14,factcols)
```

```
hr_analytics[,(factcols):=lapply(.SD,factor),.SDcols=factcols]
hr_analytics[,(numcols):=lapply(.SD,as.numeric),.SDcols=numcols]
```

```
str(hr_analytics)
```

```
## Classes 'data.table' and 'data.frame':   54808 obs. of  14 variables:
##  $ employee_id         : Factor w/ 54808 levels "1","2","4","5",..: 45806 45594 5248 1773 34271 41227 14
220 11403 51235 20135 ...
##  $ department          : Factor w/ 9 levels "Analytics","Finance",..: 8 5 8 8 9 1 5 5 1 8 ...
##  $ region              : Factor w/ 34 levels "region_1","region_10",..: 32 15 11 16 19 12 13 28 13 1 ...
##  $ education           : Factor w/ 3 levels "Bachelor's","Below Secondary",..: 3 1 1 1 1 1 1 3 1 3 ...
##  $ gender              : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 2 1 1 1 ...
##  $ recruitment_channel : Factor w/ 3 levels "other","referred",..: 3 1 3 1 1 3 1 3 1 3 ...
##  $ no_of_trainings     : Factor w/ 10 levels "1","2","3","4",..: 1 1 1 2 1 2 1 1 1 1 ...
##  $ age                 : num  35 30 34 39 45 31 31 33 28 32 ...
##  $ previous_year_rating: Factor w/ 6 levels "0","1","2","3",..: 6 6 4 2 4 4 4 4 5 6 ...
##  $ length_of_service   : num  8 4 7 10 2 7 5 6 5 5 ...
##  $ KPIs_met..80.       : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 2 ...
##  $ awards_won.         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ avg_training_score  : num  49 60 50 50 73 85 59 63 83 54 ...
##  $ is_promoted         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Seperating categorical and numerical columns for further analysis

```
cat_hr_analytics<-hr_analytics[,factcols,with=FALSE]
str(cat_hr_analytics)
```

```
## Classes 'data.table' and 'data.frame':   54808 obs. of  11 variables:
##  $ employee_id         : Factor w/ 54808 levels "1","2","4","5",..: 45806 45594 5248 1773 34271 41227 14
220 11403 51235 20135 ...
##  $ department          : Factor w/ 9 levels "Analytics","Finance",..: 8 5 8 8 9 1 5 5 1 8 ...
##  $ region              : Factor w/ 34 levels "region_1","region_10",..: 32 15 11 16 19 12 13 28 13 1 ...
##  $ education           : Factor w/ 3 levels "Bachelor's","Below Secondary",..: 3 1 1 1 1 1 1 3 1 3 ...
##  $ gender              : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 2 1 1 1 ...
##  $ recruitment_channel : Factor w/ 3 levels "other","referred",..: 3 1 3 1 1 3 1 3 1 3 ...
##  $ no_of_trainings     : Factor w/ 10 levels "1","2","3","4",..: 1 1 1 2 1 2 1 1 1 1 ...
##  $ previous_year_rating: Factor w/ 6 levels "0","1","2","3",..: 6 6 4 2 4 4 4 4 5 6 ...
##  $ KPIs_met..80.       : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 2 ...
##  $ awards_won.         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ is_promoted         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

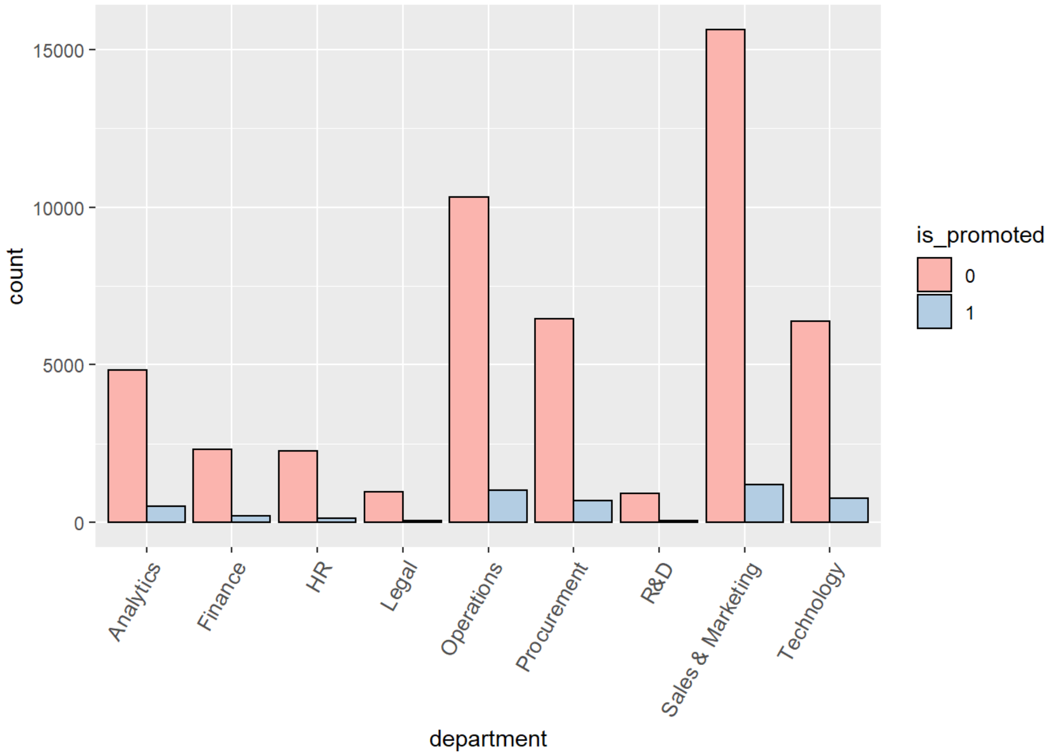
```
num_hr_analytics<-hr_analytics[,numcols,with=FALSE]
```

Analyzing Categorical Variables

Department :

#####We observe that even though Sales & Marketing department is big, employees recommended are very few, in other departments like Analytics, Operations, Technology, Procurement employee recommendation is relatively good

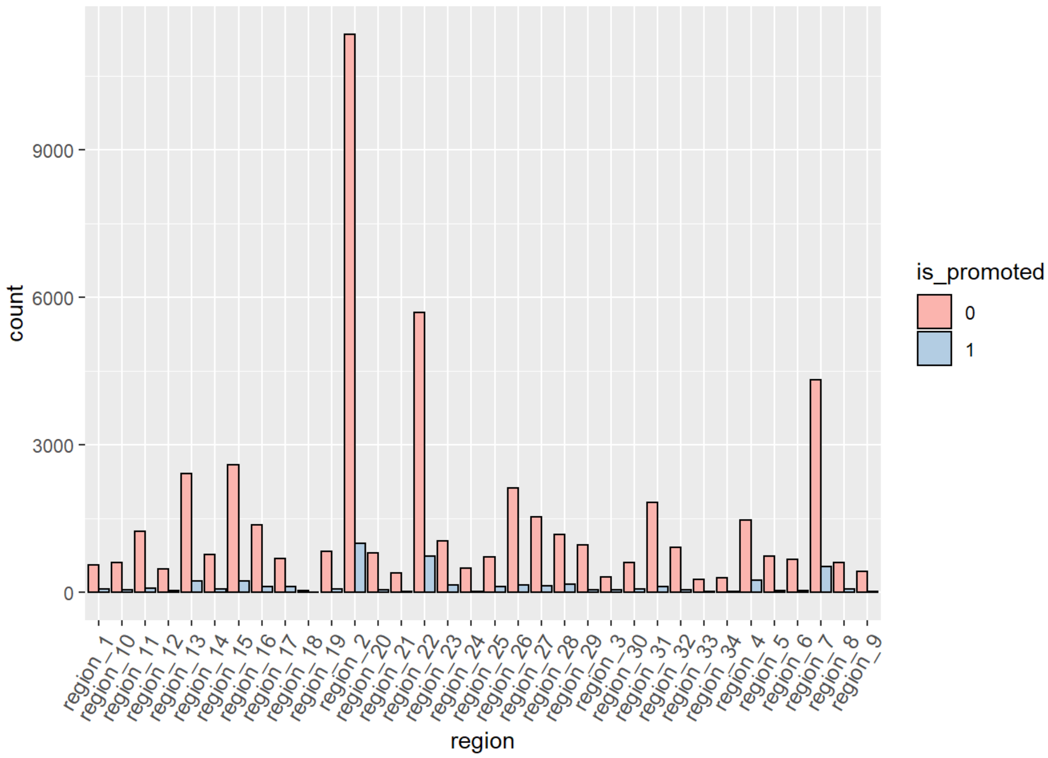
```
 ggplot(cat_hr_analytics,aes(x=department,fill=is_promoted))+
   geom_bar(position = "dodge",  color="black")+
   scale_fill_brewer(palette = "Pastel1")+
   theme(axis.text.x =element_text(angle  = 60,hjust = 1,size=10))
```



Region:

Region 7,22,19 have high recommendation for promotions

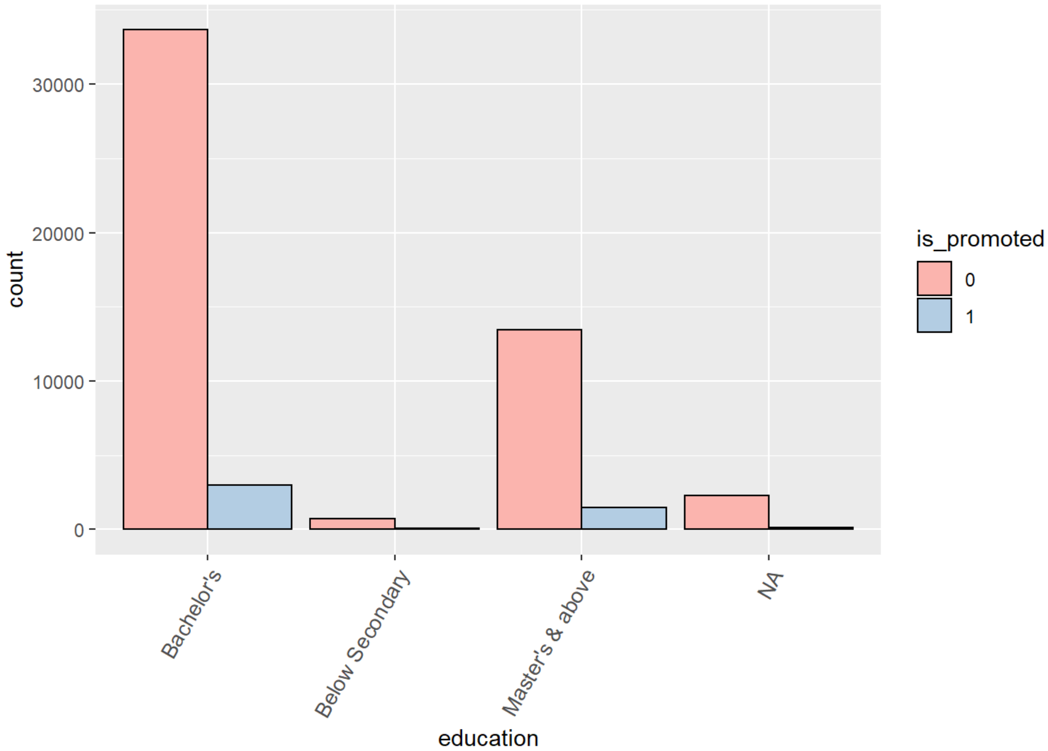
```
ggplot(cat_hr_analytics,aes(x=region,fill=is_promoted))+
  geom_bar(position = "dodge",  color="black")+
  scale_fill_brewer(palette = "Pastel1")+
  theme(axis.text.x =element_text(angle  = 60,hjust = 1,size=10))
```



Education:

People who are recommended for promotion mostly hold a Bachelor’s Degree

```
ggplot(cat_hr_analytics,aes(x=education,fill=is_promoted))+
  geom_bar(position = "dodge",  color="black")+
  scale_fill_brewer(palette = "Pastel1")+
  theme(axis.text.x =element_text(angle  = 60,hjust = 1,size=10))
```



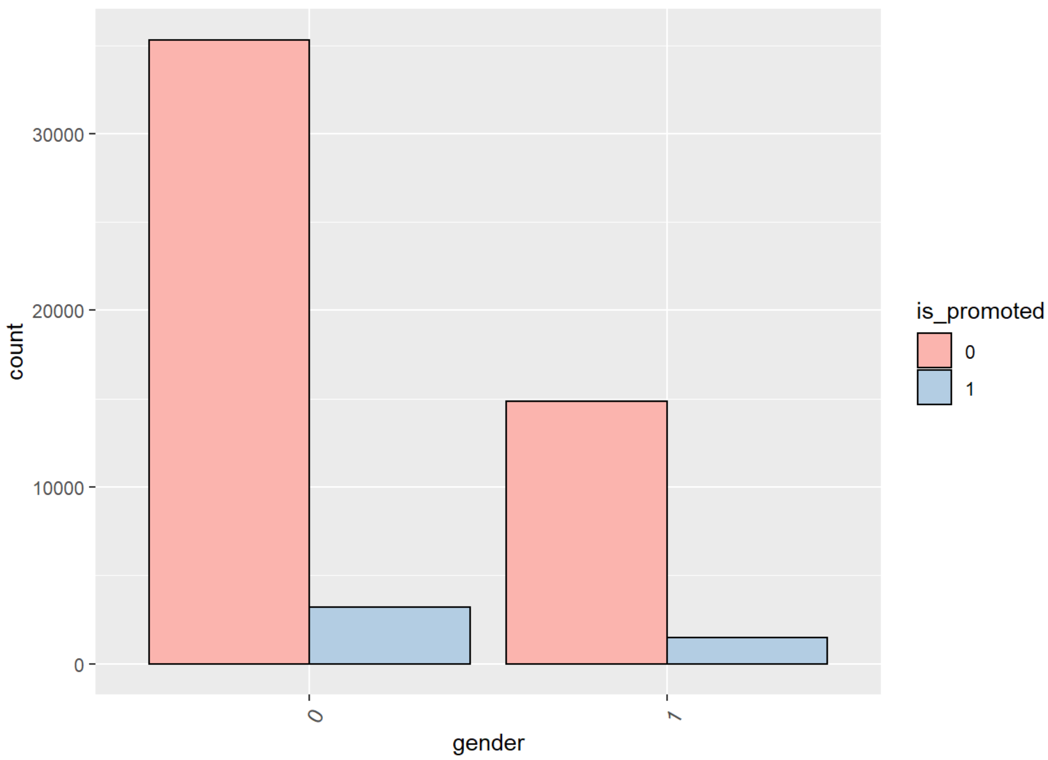
Gender:

Female data is more but rate of recommendation is less. Male data is less and rate of recommendation is high comparitively

```
prop.table(table(hr_analytics$gender,hr_analytics$is_promoted))
```

```
##
##              0          1
##   0 0.64397533 0.05840388
##   1 0.27085462 0.02676617
```

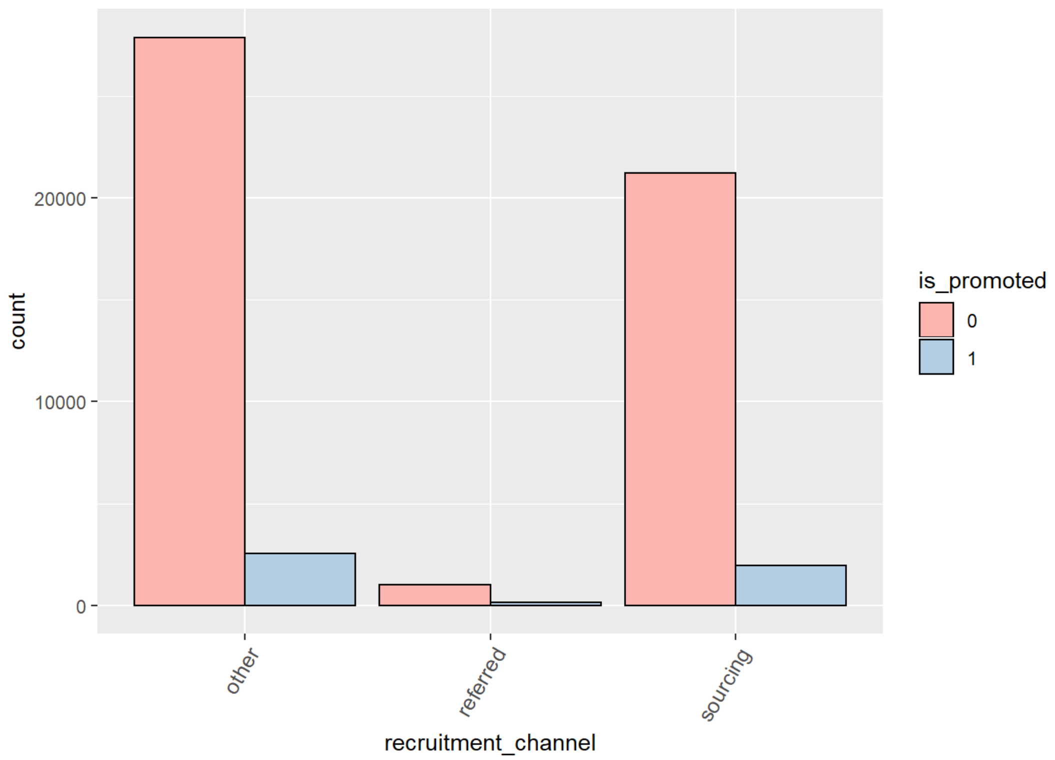
```
ggplot(cat_hr_analytics,aes(x=gender,fill=is_promoted))+
  geom_bar(position = "dodge",  color="black")+
  scale_fill_brewer(palette = "Pastel1")+
  theme(axis.text.x =element_text(angle  = 60,hjust = 1,size=10))
```



Recruitment Channel :

Employees recruited from other channel have higher probability of being recommended for promotion

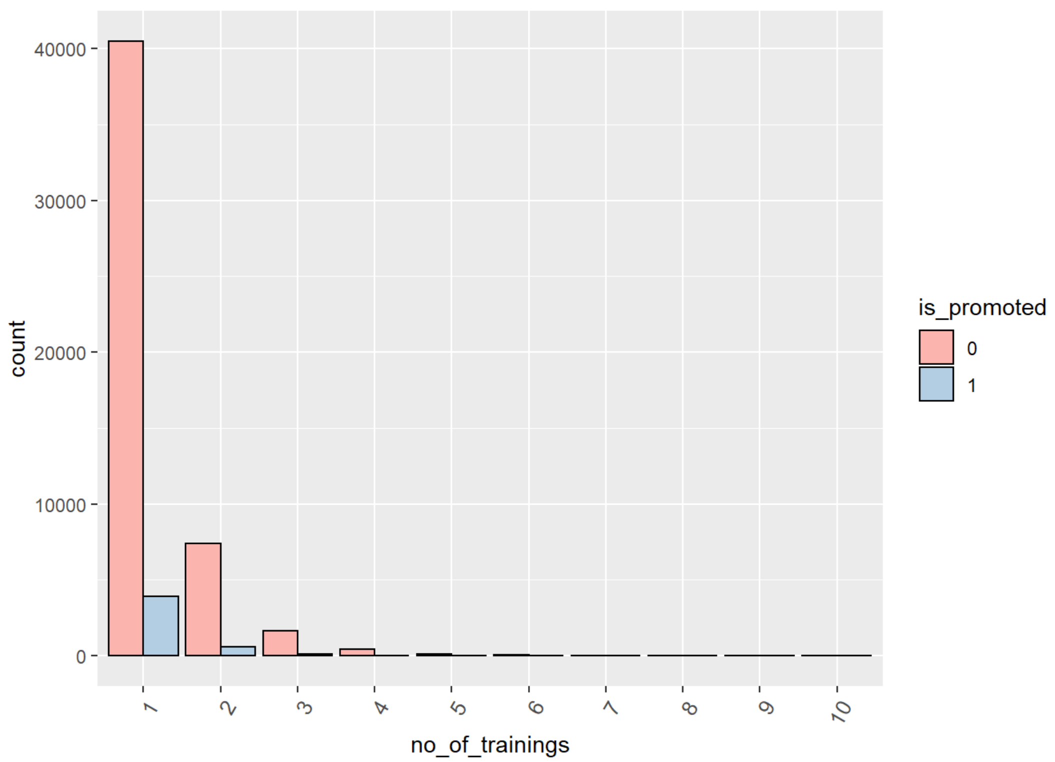
```
ggplot(cat_hr_analytics,aes(x=recruitment_channel,fill=is_promoted))+
  geom_bar(position = "dodge",  color="black")+
  scale_fill_brewer(palette = "Pastel1")+
  theme(axis.text.x =element_text(angle  = 60,hjust = 1,size=10))
```



Number of trainings:

It doesn't seem to add much value to the recommendation

```
ggplot(cat_hr_analytics,aes(x=no_of_trainings,fill=is_promoted))+
  geom_bar(position = "dodge",  color="black")+
  scale_fill_brewer(palette = "Pastel1")+
  theme(axis.text.x =element_text(angle  = 60,hjust = 1,size=10))
```



Age:

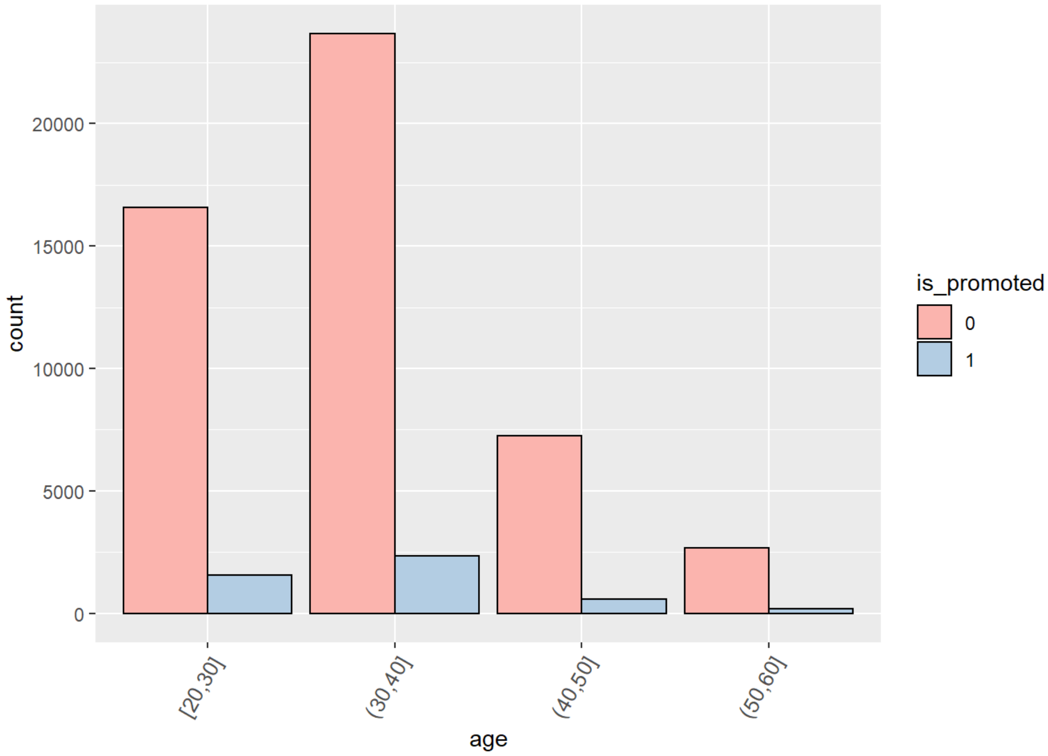
Binned age variable 20-30 31-40 41-50 51-60

```
#num_hr_analytics[,age:=hr_analytics$age]
#str(num_hr_analytics)
num_hr_analytics[,age:=cut(x=age,breaks=c(20,30,40,50,60),include.lowest = TRUE)]
num_hr_analytics[,age:=factor(age)]
unique(num_hr_analytics$age)
```

```
## [1] (30,40] [20,30] (40,50] (50,60]
## Levels: [20,30] (30,40] (40,50] (50,60]
```

```
num_hr_analytics$is_promoted<-hr_analytics$is_promoted
```

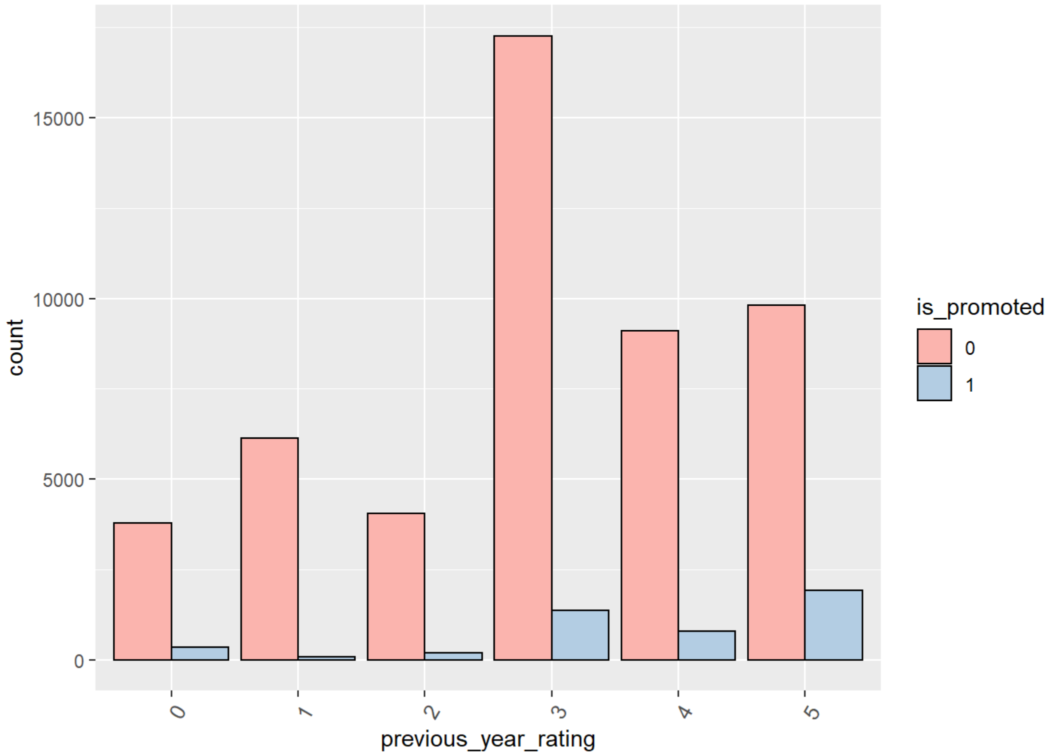
```
  ggplot(num_hr_analytics,aes(x=age,fill=is_promoted))+
   geom_bar(position = "dodge",  color="black")+
   scale_fill_brewer(palette = "Pastel1")+
   theme(axis.text.x =element_text(angle  = 60,hjust = 1,size=10))
```



Previous Year Rating :

Employees with previous year rating of 5 have fair amount of chance of being recommended for promotion

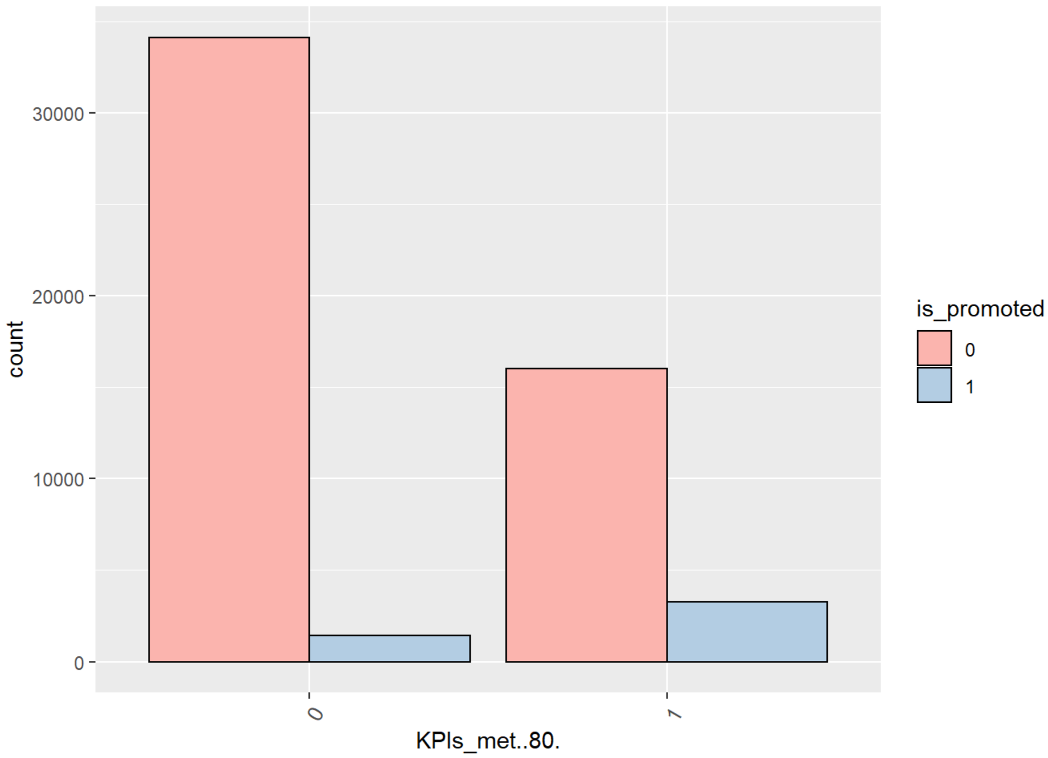
```
 ggplot(cat_hr_analytics,aes(x=previous_year_rating,fill=is_promoted))+
   geom_bar(position = "dodge",  color="black")+
   scale_fill_brewer(palette = "Pastel1")+
   theme(axis.text.x =element_text(angle  = 60,hjust = 1,size=10))
```



KPI's met >80%:

Employees who have KPI's greater than 80 have higher chances of being recommended for promotion

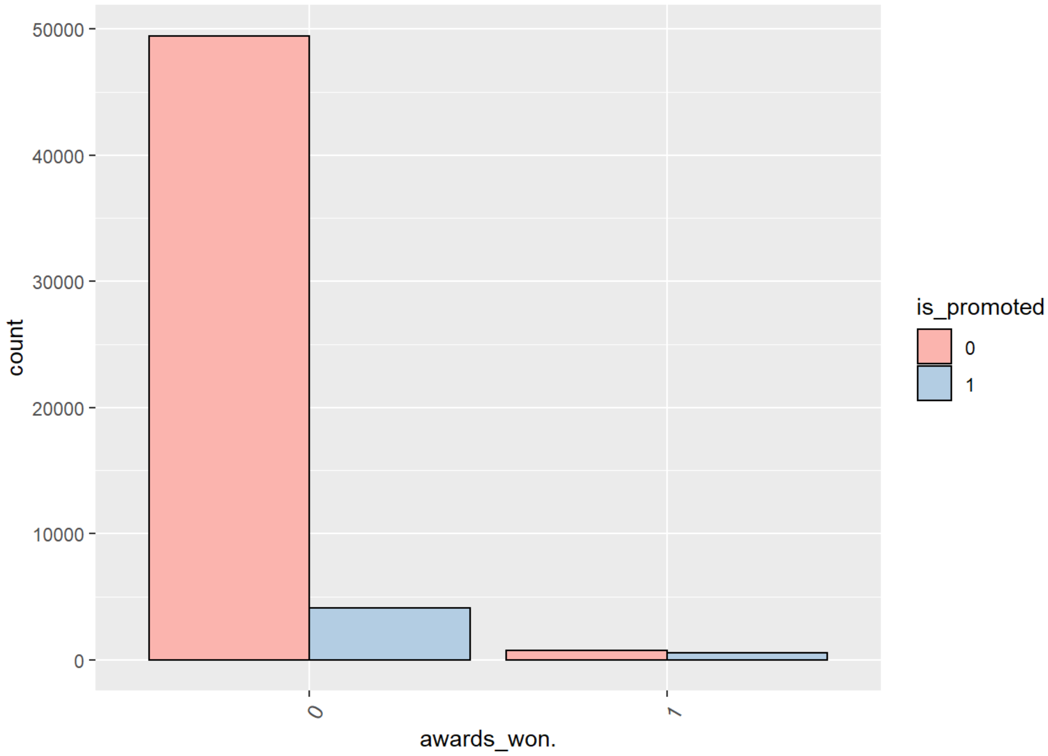
```
ggplot(cat_hr_analytics,aes(x=KPIs_met..80.,fill=is_promoted))+
  geom_bar(position = "dodge",  color="black")+
  scale_fill_brewer(palette = "Pastel1")+
  theme(axis.text.x =element_text(angle  = 60,hjust = 1,size=10))
```



Awards Won:

People who have not won more awards are not likely to be recommended for promotion

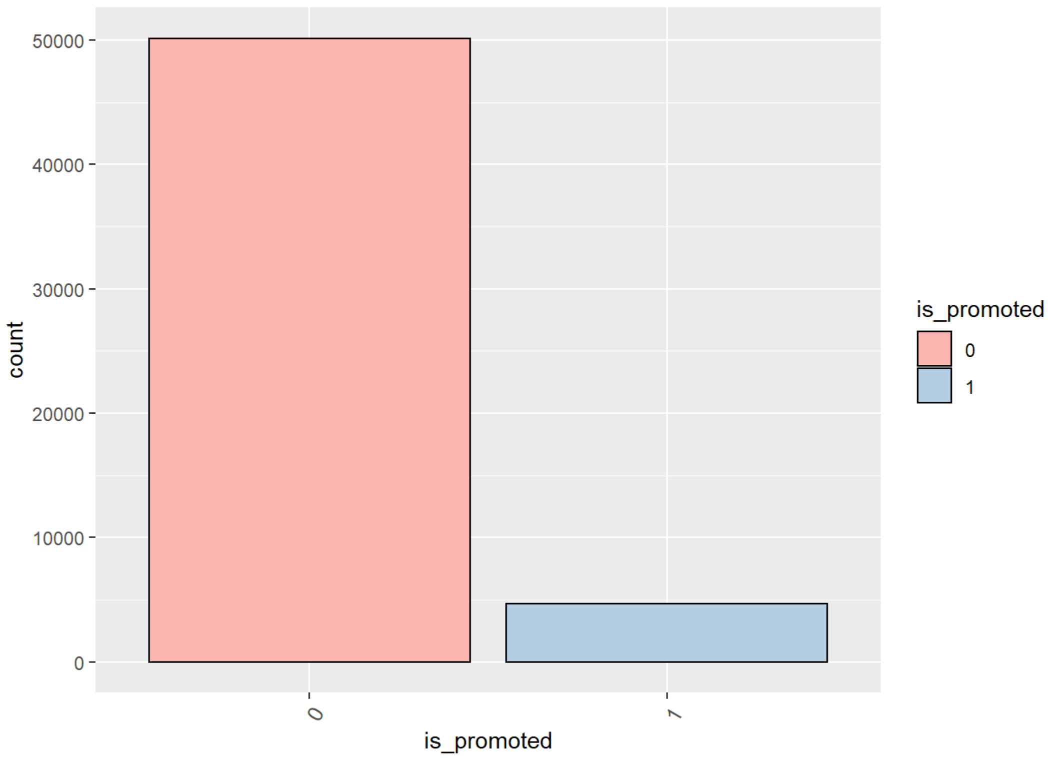
```
ggplot(cat_hr_analytics,aes(x=awards_won.,fill=is_promoted))+
  geom_bar(position = "dodge",  color="black")+
  scale_fill_brewer(palette = "Pastel1")+
  theme(axis.text.x =element_text(angle  = 60,hjust = 1,size=10))
```



is_promoted:

This indicates an huge imbalance in the target variable for classification. This issue needs to be addressed before modeling.

```
ggplot(cat_hr_analytics,aes(x=is_promoted,fill=is_promoted))+
  geom_bar(position = "dodge",  color="black")+
  scale_fill_brewer(palette = "Pastel1")+
  theme(axis.text.x =element_text(angle  = 60,hjust = 1,size=10))
```




```
prop.table(table(cat_hr_analytics$is_promoted))
```

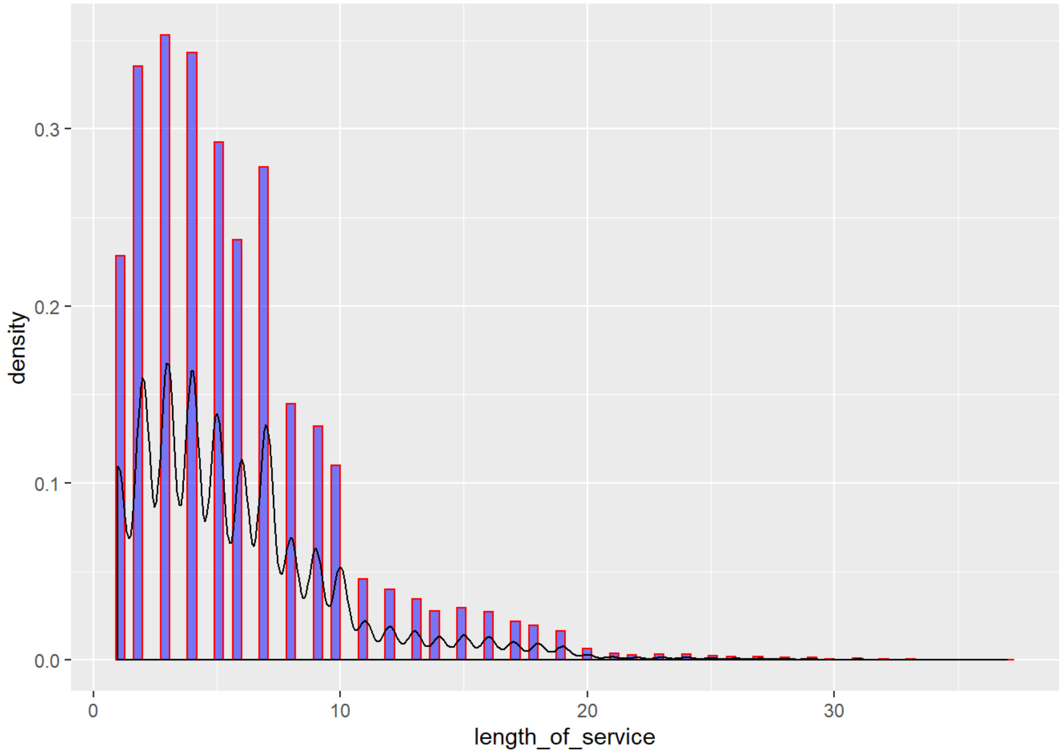
```
##
##          0          1
## 0.91482995 0.08517005
```

Exploring Numerical Variables

Length_of_Service:

Distribution shows a right skewness

```
ggplot(data = num_hr_analytics, aes(x= length_of_service, y=..density..)) +
geom_histogram(fill="blue",color="red",alpha = 0.5,bins =100) +
geom_density()
```

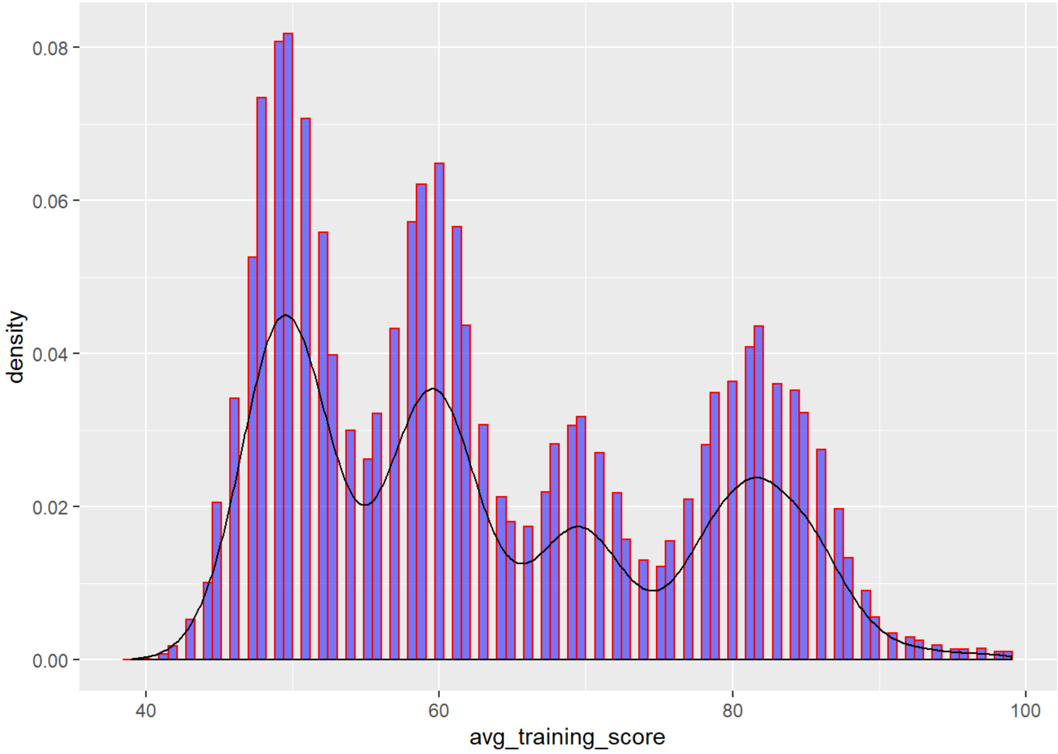



```
ggplotly()
```

Avg_Training_Score:

Skewness is not evident in the distribution

```
ggplot(data = num_hr_analytics, aes(x= avg_training_score, y=..density..)) +
 geom_histogram(fill="blue",color="red",alpha = 0.5,bins =100) +
 geom_density()
```



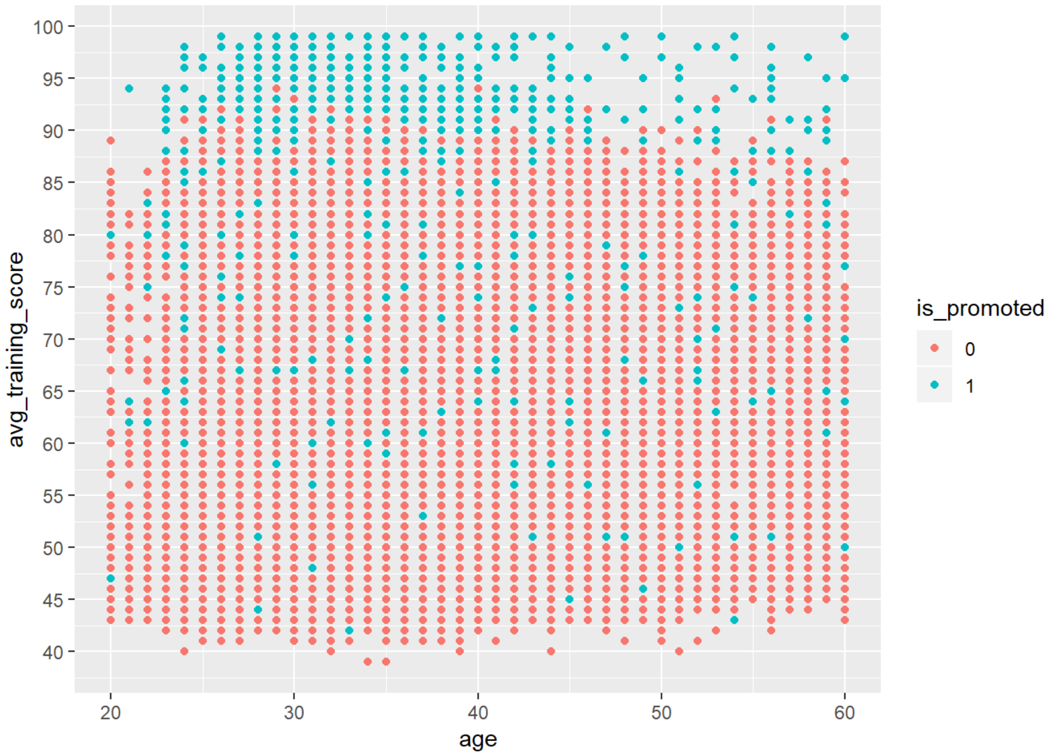

```
ggplotly()
```

Avg_training_score vs Age:

Higher the average score across all ages, are highly recommended for promotion

```
num_hr_analytics[,age:=NULL]
num_hr_analytics[,age:=hr_analytics$age]
```

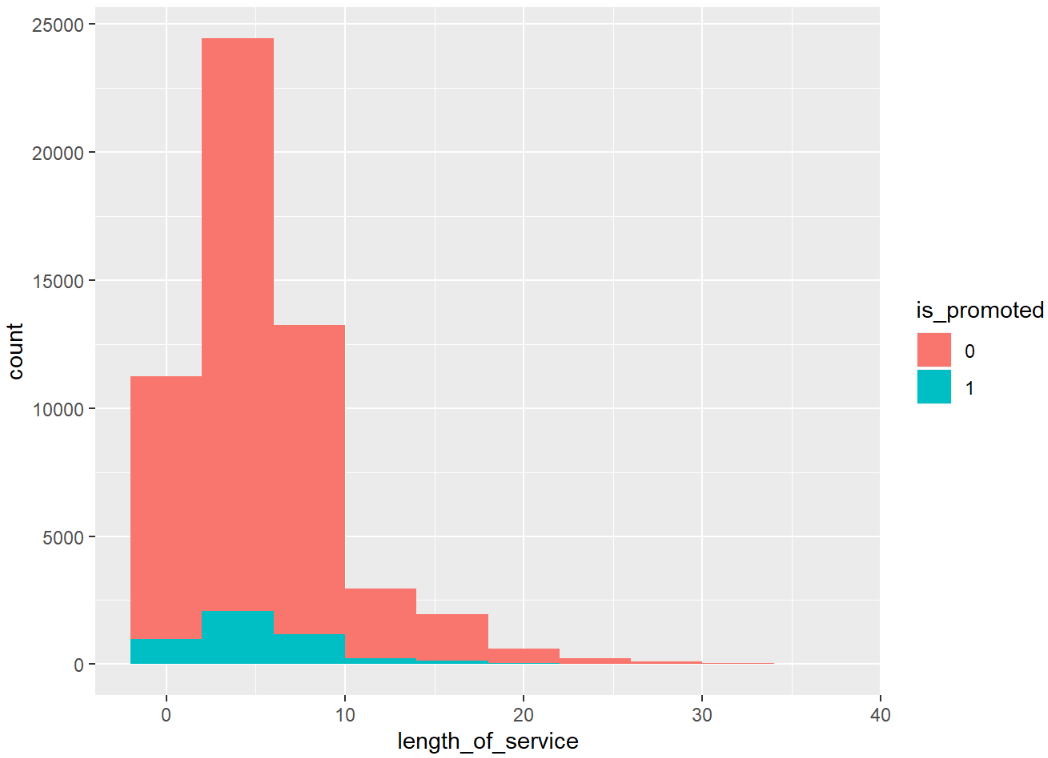
```r
# create scatter plot
ggplot(data=num_hr_analytics,aes(x=age,y=avg_training_score))+geom_point(aes(colour=is_promoted))+scale_y_co
ntinuous("avg_training_score",breaks = seq(0,100,5))
```



Length of Services vs Recommendation :

Employees whose length of service is in the range of 1 to 10 years are highly recommended for promotion

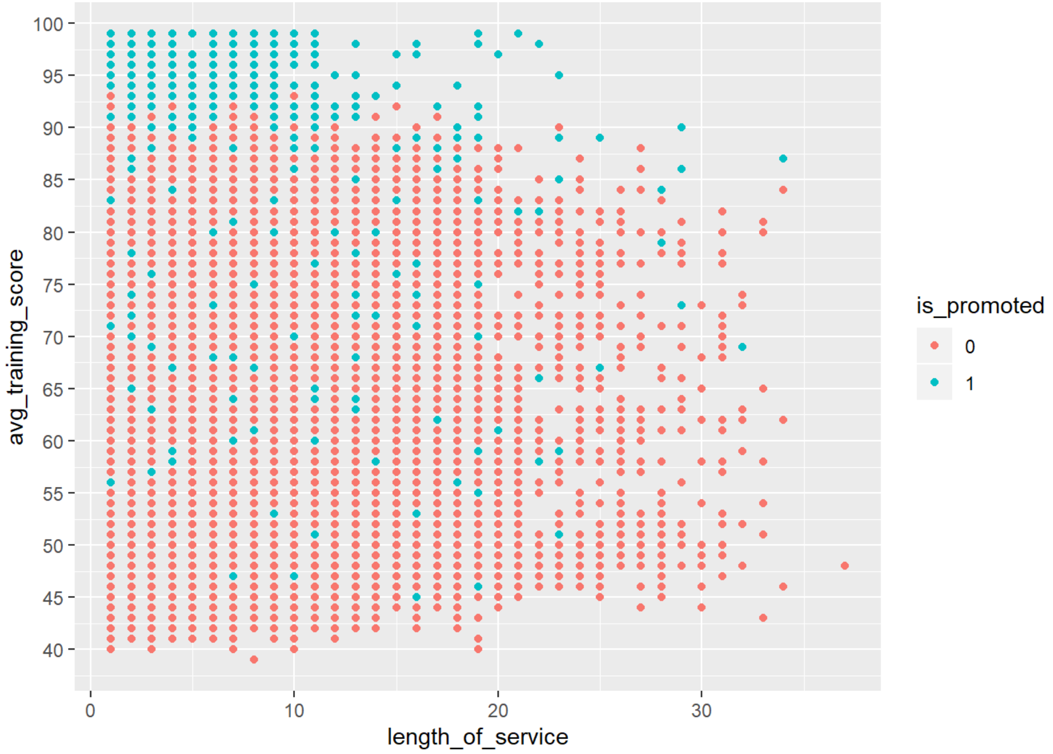
```r
ggplot(data=num_hr_analytics,aes(x=length_of_service,fill=is_promoted))+geom_histogram(bins=10)
```



Length of Service vs Avg Training Score :

Employees with length of service of 1 to 11 years along with high average training scores are highly recommended for promotion

```
# create scatter plot
ggplot(data=num_hr_analytics,aes(x=length_of_service,y=avg_training_score))+geom_point(aes(colour=is_promote
d))+scale_y_continuous("avg_training_score",breaks = seq(0,100,5))
```



Removing reduntant is_promoted and age from numerical data

```
num_hr_analytics[,age:=NULL]
num_hr_analytics[,is_promoted:=NULL]
```

```
str(num_hr_analytics)
```

```
## Classes 'data.table' and 'data.frame':   54808 obs. of  2 variables:
##  $ length_of_service : num  8 4 7 10 2 7 5 6 5 5 ...
##  $ avg_training_score: num  49 60 50 50 73 85 59 63 83 54 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Adding age back to categorical data

```
cat_hr_analytics[,age:=hr_analytics$age]
cat_hr_analytics[,age:=cut(x=age,breaks=c(20,30,40,50,60),include.lowest = TRUE)]
cat_hr_analytics[,age:=factor(age)]
unique(cat_hr_analytics$age)
```

```
## [1] (30,40] [20,30] (40,50] (50,60]
## Levels: [20,30] (30,40] (40,50] (50,60]
```

```
cat_hr_analytics[,no_of_trainings:=NULL]
num_hr_analytics[,no_of_trainings:=as.numeric(hr_analytics$no_of_trainings)]
```

```
str(cat_hr_analytics)
```

```
## Classes 'data.table' and 'data.frame':   54808 obs. of  11 variables:
##  $ employee_id         : Factor w/ 54808 levels "1","2","4","5",..: 45806 45594 5248 1773 34271 41227 14
220 11403 51235 20135 ...
##  $ department          : Factor w/ 9 levels "Analytics","Finance",..: 8 5 8 8 9 1 5 5 1 8 ...
##  $ region              : Factor w/ 34 levels "region_1","region_10",..: 32 15 11 16 19 12 13 28 13 1 ...
##  $ education           : Factor w/ 3 levels "Bachelor's","Below Secondary",..: 3 1 1 1 1 1 1 3 1 3 ...
##  $ gender              : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 2 1 1 1 ...
##  $ recruitment_channel : Factor w/ 3 levels "other","referred",..: 3 1 3 1 1 3 1 3 1 3 ...
##  $ previous_year_rating: Factor w/ 6 levels "0","1","2","3",..: 6 6 4 2 4 4 4 4 5 6 ...
##  $ KPIs_met..80.       : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 2 ...
##  $ awards_won.         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ is_promoted         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ age                 : Factor w/ 4 levels "[20,30]","(30,40]",..: 2 1 2 2 3 2 2 2 1 2 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Replacing NA's with 'Unavailable' before applying Models

```
# Convert to characters
cat_hr_analytics<-cat_hr_analytics[,names(cat_hr_analytics):=lapply(.SD,as.character),.SDcols=names(cat_hr_a
nalytics)]
for( i in names(cat_hr_analytics))
{

  if(length(which(is.na(cat_hr_analytics[[i]]))>0))
  {
    cat_hr_analytics[[i]][is.na(cat_hr_analytics[[i]])]<-'Unavailable'
  }

}
# convert back to factors
cat_hr_analytics<-cat_hr_analytics[,names(cat_hr_analytics):=lapply(.SD,factor),.SDcols=names(cat_hr_analyti
cs)]
grep('NA',cat_hr_analytics)
```

```
## integer(0)
```

```
str(num_hr_analytics)
```

```
## Classes 'data.table' and 'data.frame':   54808 obs. of  3 variables:
##  $ length_of_service : num  8 4 7 10 2 7 5 6 5 5 ...
##  $ avg_training_score: num  49 60 50 50 73 85 59 63 83 54 ...
##  $ no_of_trainings   : num  1 1 1 2 1 2 1 1 1 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Machine Learning

```
rm(hr_analytics)
```

Combine numerical and categorical data

```
hr_analytics<-cbind(cat_hr_analytics,num_hr_analytics)
```

```
unique(hr_analytics$is_promoted)
```

```
## [1] 0 1
## Levels: 0 1
```

```
str(hr_analytics)
```

```
## Classes 'data.table' and 'data.frame':   54808 obs. of  14 variables:
##  $ employee_id         : Factor w/ 54808 levels "1","10","100",..: 43141 42907 50654 11929 30338 38057 8
054 4923 49174 14624 ...
##  $ department          : Factor w/ 9 levels "Analytics","Finance",..: 8 5 8 8 9 1 5 5 1 8 ...
##  $ region              : Factor w/ 34 levels "region_1","region_10",..: 32 15 11 16 19 12 13 28 13 1 ...
##  $ education           : Factor w/ 4 levels "Bachelor's","Below Secondary",..: 3 1 1 1 1 1 1 3 1 3 ...
##  $ gender              : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 2 1 1 1 ...
##  $ recruitment_channel : Factor w/ 3 levels "other","referred",..: 3 1 3 1 1 3 1 3 1 3 ...
##  $ previous_year_rating: Factor w/ 6 levels "0","1","2","3",..: 6 6 4 2 4 4 4 4 5 6 ...
##  $ KPIs_met..80.       : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 2 ...
##  $ awards_won.         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ is_promoted         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ age                 : Factor w/ 4 levels "(30,40]","(40,50]",..: 1 4 1 1 2 1 1 1 4 1 ...
##  $ length_of_service   : num  8 4 7 10 2 7 5 6 5 5 ...
##  $ avg_training_score  : num  49 60 50 50 73 85 59 63 83 54 ...
##  $ no_of_trainings     : num  1 1 1 2 1 2 1 1 1 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Making train and test data

```
# Random sample indexes
train_index = sample(1:nrow(hr_analytics), 0.75 * nrow(hr_analytics))
test_index= setdiff(1:nrow(hr_analytics), train_index)
# Build train and test sets
train_set = hr_analytics[train_index, ]
test_set = hr_analytics[test_index, ]
setDF(train_set)
setDF(test_set)
```

```
str(train_set)
```

```
## 'data.frame':    41106 obs. of  14 variables:
##  $ employee_id         : Factor w/ 54808 levels "1","10","100",..: 36791 48166 4995 5462 7254 34039 4856
2 44052 28456 7669 ...
##  $ department          : Factor w/ 9 levels "Analytics","Finance",..: 8 8 6 8 8 3 8 5 5 6 ...
##  $ region              : Factor w/ 34 levels "region_1","region_10",..: 12 12 6 8 1 12 3 31 13 8 ...
##  $ education           : Factor w/ 4 levels "Bachelor's","Below Secondary",..: 3 3 3 3 1 2 1 1 1 1 ...
##  $ gender              : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 1 1 2 2 ...
##  $ recruitment_channel : Factor w/ 3 levels "other","referred",..: 1 1 3 1 3 3 3 3 3 3 ...
##  $ previous_year_rating: Factor w/ 6 levels "0","1","2","3",..: 6 6 6 4 4 1 6 4 5 4 ...
##  $ KPIs_met..80.       : Factor w/ 2 levels "0","1": 2 2 1 1 2 1 1 1 1 1 ...
##  $ awards_won.         : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 1 ...
##  $ is_promoted         : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 1 ...
##  $ age                 : Factor w/ 4 levels "(30,40]","(40,50]",..: 1 1 1 2 4 4 4 4 4 4 ...
##  $ length_of_service   : num  6 5 8 7 3 1 3 3 6 8 ...
##  $ avg_training_score  : num  49 97 69 49 47 49 45 56 62 69 ...
##  $ no_of_trainings     : num  1 1 1 1 1 1 2 1 1 1 ...
```

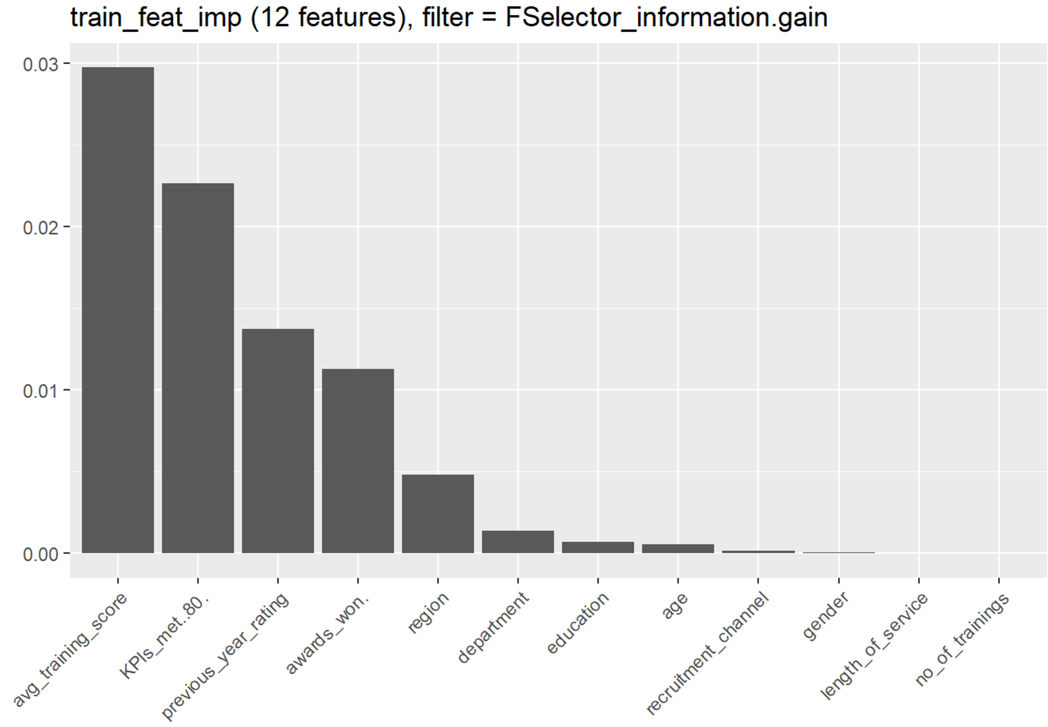
```
train_feat_imp<-train_set[,-1]
setDF(train_feat_imp)
test_feat_imp<-test_set[,-1]
setDF(test_feat_imp)
train.task <- makeClassifTask(data = train_feat_imp,target = "is_promoted")
test.task <- makeClassifTask(data=test_feat_imp,target = "is_promoted")

levels(test_feat_imp$no_of_trainings)
```

```
## NULL
```

Variable Importance Chart before applying models on the data

```
# get variable importance chart
var_imp<-generateFilterValuesData(train.task,method=c('FSelector_information.gain'))
plotFilterValues(var_imp,feat.type.cols=FALSE)
```



Handling Imbalanced Data

ROSE : Over-Sampling increases the number of instances in the minority class by randomly replicating them in order to present a higher representation of the minority class in the sample.

```
data.rose<-ROSE(is_promoted~.,data=train_feat_imp,seed=1)$data
table(data.rose$is_promoted)
```

```
##
##     0     1
## 20602 20504
```

Recursive Partitioning(rpart) with ROSE data

```
tree.both<-rpart(is_promoted~.,data=data.rose)
```

```
pred.tree.rose<-predict(tree.both,newdata=test_feat_imp,type='class')
```

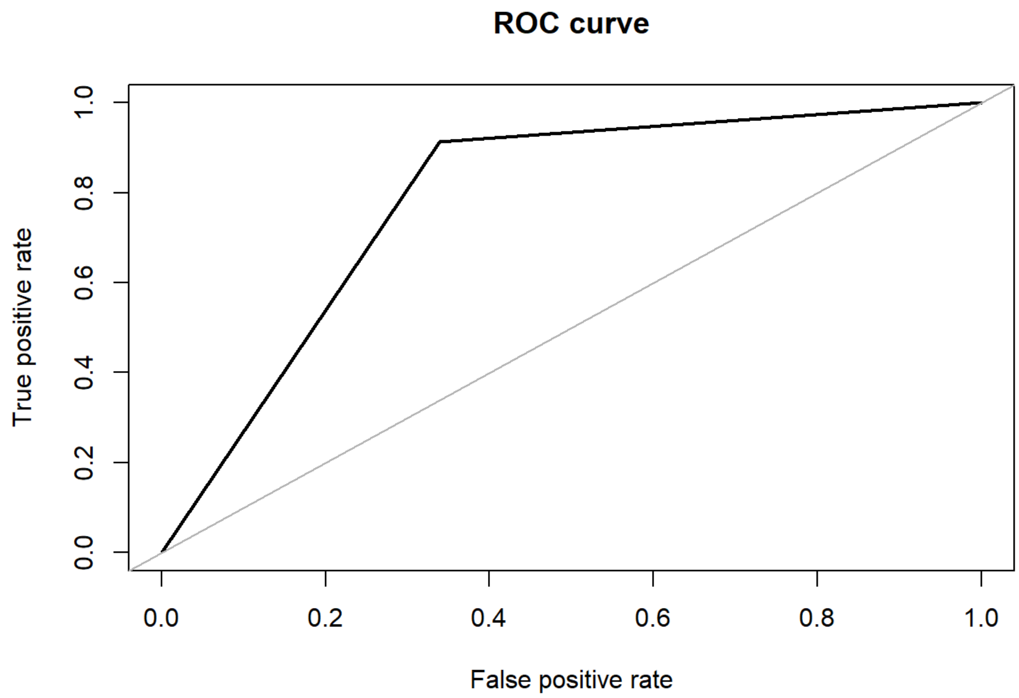
```
confmat.tree.rose<-table(pred.tree.rose,test_feat_imp$is_promoted)
```

Accuracy of the Recursive partioning of ROSE data

```
accuracy.tree.rose<-sum(diag(confmat.tree.rose))/sum(confmat.tree.rose)
```

AUC of the predicted data

```
roc.curve(test_feat_imp$is_promoted,pred.tree.rose)
```




```
## Area under the curve (AUC): 0.787
```

Random Forest using ROSE

```
rfrose<-randomForest(is_promoted ~., data=data.rose,importance=TRUE)
```

Fine tuning parameters of Random Forest model

```
rfrosetune1<-randomForest(is_promoted ~., data=data.rose,ntree=500,mtry=6,importance=TRUE)
```

```
# Predicting on train set
predTrain.rose<-predict(rfrosetune1,data.rose,type='class')
# Checking classification accuracy
table(predTrain.rose,data.rose$is_promoted)
```

```
##
## predTrain.rose     0     1
##              0 20602     0
##              1     0 20504
```

```
# Predicting on validation set
predValid.rose<-predict(rfrosetune1,test_feat_imp,type='class')
# Checking classification accuracy
table(predValid.rose,test_feat_imp$is_promoted)
```

```
##
## predValid.rose     0     1
##              0 10352   275
##              1  2224   851
```

Accuracy of the Random Forest for ROSE data

```
mean(predValid.rose==test_feat_imp$is_promoted)
```

```
## [1] 0.8176179
```

Confusion Matrix

```
confmat.rf.rose<-table(predValid.rose,test_feat_imp$is_promoted)
```

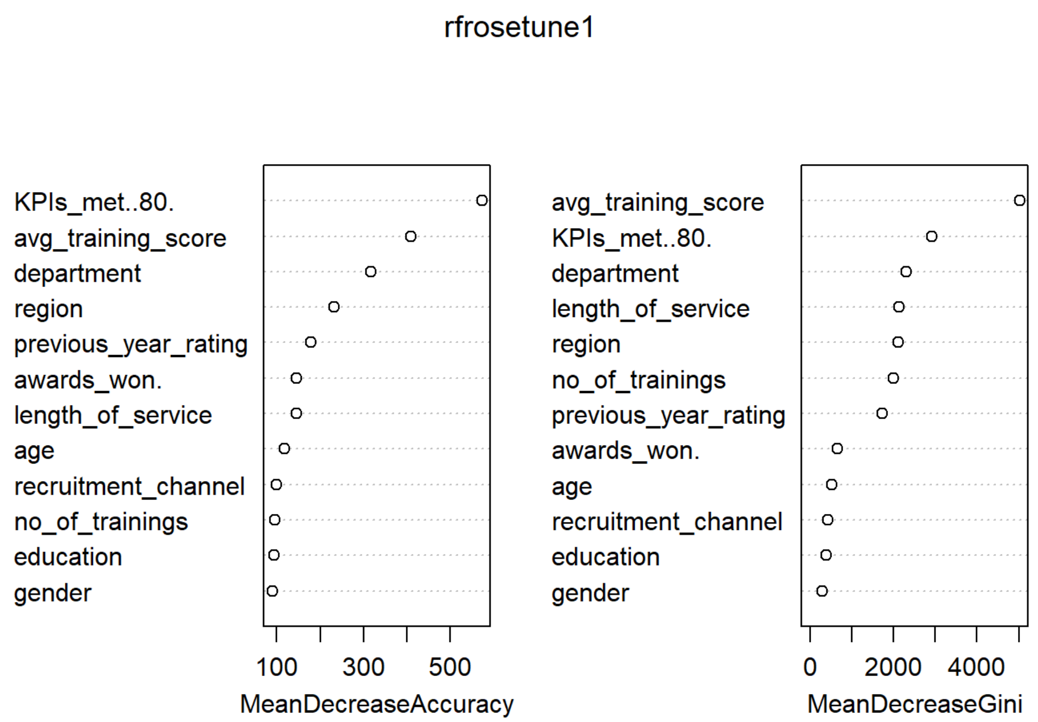
Important variables obtained after applying Random forest

```
importance(rfrosetune1)
```

```
##                               0         1 MeanDecreaseAccuracy
## department            247.40855  93.09987            316.11418
## region                 76.71298 230.35020            231.18114
## education              19.82315  94.30782             94.40395
## gender                 28.08425  88.01132             89.16483
## recruitment_channel    34.35670 100.12472            100.01415
## previous_year_rating 173.90658 141.25385            177.79359
## KPIs_met..80.         353.06340 447.53236            572.82743
## awards_won.           129.28654 103.00253            145.73126
## age                    23.27560 116.04304            117.69499
## length_of_service      41.84487 143.93381            145.22317
## avg_training_score    304.28998 208.81698            409.70370
## no_of_trainings        30.13959  98.47853             95.06337
##                      MeanDecreaseGini
## department                  2310.2197
## region                      2107.1767
## education                    392.3694
## gender                       294.1740
## recruitment_channel          425.1816
## previous_year_rating        1733.5208
## KPIs_met..80.               2928.8100
## awards_won.                  663.6238
## age                          518.8520
## length_of_service           2130.7469
## avg_training_score          5020.2257
## no_of_trainings             2006.4979
```

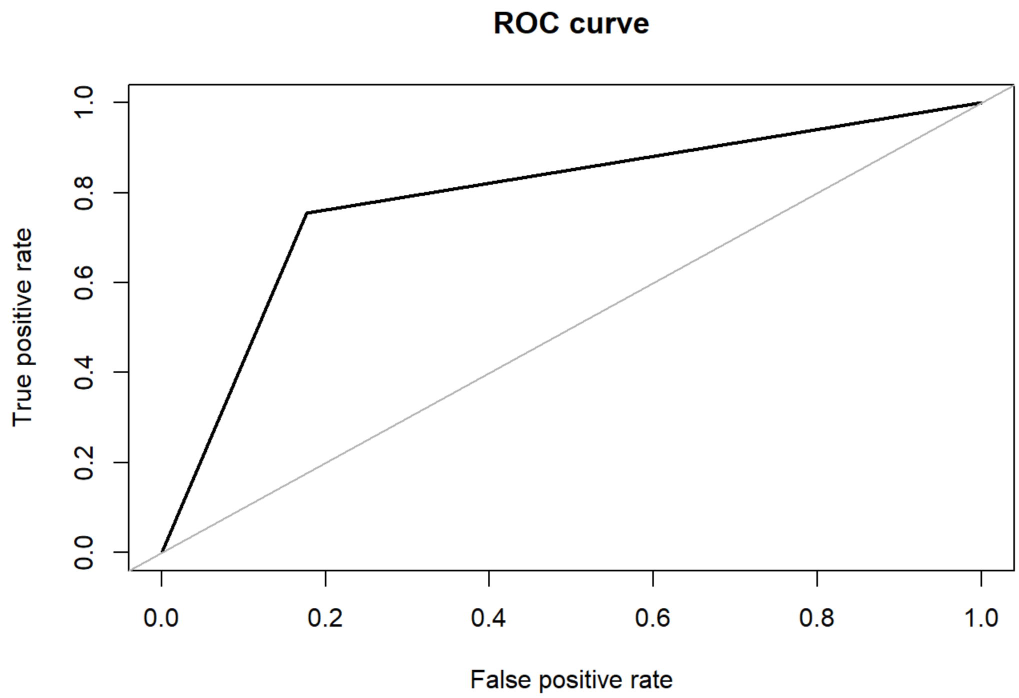
Variable Importance plot

```
varImpPlot(rfrosetune1)
```



AUC for Random Forest

```
roc.curve(test_feat_imp$is_promoted,predValid.rose)
```




```
## Area under the curve (AUC): 0.789
```

Logistic Regression using ROSE

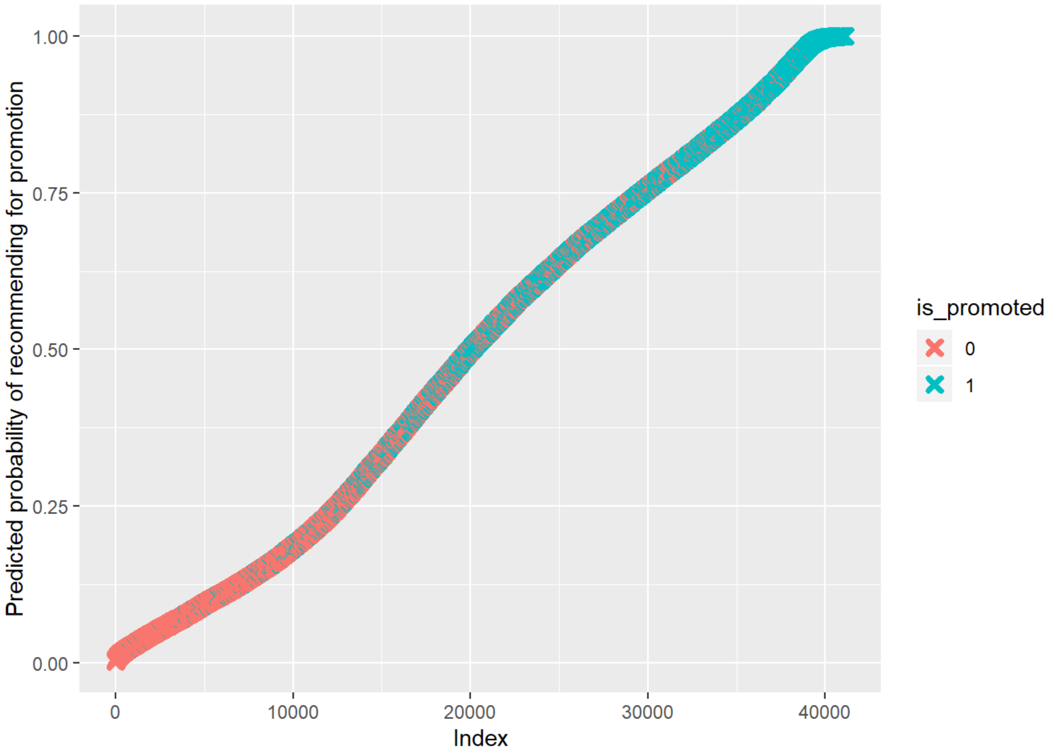
```
logistic_regres <- glm( is_promoted ~. ,data=data.rose, family="binomial")
summary(logistic_regres)
```

```
##
## Call:
## glm(formula = is_promoted ~ ., family = "binomial", data = data.rose)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0934  -0.6831  -0.1132   0.7933   2.7446
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -14.782248   0.251768 -58.714  < 2e-16 ***
## departmentFinance            3.421054   0.089321  38.301  < 2e-16 ***
## departmentHR                 4.732339   0.110408  42.862  < 2e-16 ***
## departmentLegal              2.945706   0.123149  23.920  < 2e-16 ***
## departmentOperations         3.611650   0.073894  48.876  < 2e-16 ***
## departmentProcurement        2.069565   0.063005  32.847  < 2e-16 ***
## departmentR&D               -0.711493   0.107829  -6.598 4.16e-11 ***
## departmentSales & Marketing  5.281118   0.092171  57.297  < 2e-16 ***
## departmentTechnology         0.934929   0.054140  17.269  < 2e-16 ***
## regionregion_10              0.073187   0.165215   0.443 0.657778
## regionregion_11             -0.335020   0.148715  -2.253 0.024274 *
## regionregion_12             -0.904219   0.203752  -4.438 9.09e-06 ***
## regionregion_13             -0.009409   0.133024  -0.071 0.943613
## regionregion_14              0.121172   0.161176   0.752 0.452174
## regionregion_15              0.032255   0.133314   0.242 0.808824
## regionregion_16             -0.089937   0.144668  -0.622 0.534153
## regionregion_17              0.413849   0.152866   2.707 0.006784 **
## regionregion_18             -0.380812   0.844853  -0.451 0.652174
## regionregion_19             -0.285929   0.159180  -1.796 0.072453 .
## regionregion_2               0.134312   0.124261   1.081 0.279751
## regionregion_20             -0.285844   0.164933  -1.733 0.083079 .
## regionregion_21             -0.490360   0.213359  -2.298 0.021545 *
## regionregion_22              0.513502   0.125751   4.083 4.44e-05 ***
## regionregion_23              0.259205   0.145241   1.785 0.074317 .
## regionregion_24              0.116953   0.187952   0.622 0.533777
## regionregion_25              0.575054   0.152554   3.770 0.000164 ***
```

```
## regionregion_26              -0.125426   0.137167  -0.914 0.360506
## regionregion_27              -0.135019   0.142477  -0.948 0.343305
## regionregion_28               0.492116   0.142121   3.463 0.000535 ***
## regionregion_29              -0.566281   0.164669  -3.439 0.000584 ***
## regionregion_3               -0.080001   0.205587  -0.389 0.697175
## regionregion_30               0.192616   0.164394   1.172 0.241329
## regionregion_31              -0.363578   0.142188  -2.557 0.010557 *
## regionregion_32              -0.718943   0.167380  -4.295 1.74e-05 ***
## regionregion_33              -0.478016   0.247735  -1.930 0.053664 .
## regionregion_34              -1.499384   0.305783  -4.903 9.42e-07 ***
## regionregion_4                0.810250   0.134939   6.005 1.92e-09 ***
## regionregion_5               -0.519974   0.174792  -2.975 0.002932 **
## regionregion_6               -0.505902   0.177455  -2.851 0.004360 **
## regionregion_7                0.397331   0.126657   3.137 0.001706 **
## regionregion_8               -0.270966   0.170976  -1.585 0.113008
## regionregion_9               -1.486924   0.251059  -5.923 3.17e-09 ***
## educationBelow Secondary     -0.111051   0.102680  -1.082 0.279462
## educationMaster's & above     0.134022   0.032092   4.176 2.96e-05 ***
## educationUnavailable         -0.571850   0.074376  -7.689 1.49e-14 ***
## gender1                      -0.034887   0.029065  -1.200 0.230024
## recruitment_channelreferred  -0.123282   0.080835  -1.525 0.127232
## recruitment_channelsourcing  -0.024828   0.025950  -0.957 0.338689
## previous_year_rating1        -1.698904   0.084980 -19.992  < 2e-16 ***
## previous_year_rating2        -0.473640   0.072828  -6.504 7.84e-11 ***
## previous_year_rating3        -0.127019   0.055325  -2.296 0.021684 *
## previous_year_rating4        -0.398688   0.058880  -6.771 1.28e-11 ***
## previous_year_rating5         0.312829   0.055870   5.599 2.15e-08 ***
## KPIs_met..80.1                2.105241   0.028943  72.738  < 2e-16 ***
## awards_won.1                  1.925614   0.074155  25.968  < 2e-16 ***
## age(40,50]                   -0.315081   0.041941  -7.512 5.80e-14 ***
## age(50,60]                   -0.698930   0.071649  -9.755  < 2e-16 ***
## age[20,30]                    0.192212   0.035044   5.485 4.14e-08 ***
## length_of_service             0.026941   0.003825   7.043 1.88e-12 ***
## avg_training_score            0.161229   0.002358  68.369  < 2e-16 ***
## no_of_trainings              -0.199907   0.023217  -8.610  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 56985  on 41105  degrees of freedom
## Residual deviance: 38130  on 41045  degrees of freedom
## AIC: 38252
##
## Number of Fisher Scoring iterations: 5
```

```
#probablity_pred
predicted.rose<-data.frame(probability.of.recommended=logistic_regres$fitted.values,is_promoted=data.rose$is
_promoted)
predicted.rose <- predicted.rose[order(predicted.rose$probability.of.recommended, decreasing=FALSE),]
predicted.rose$rank <- 1:nrow(predicted.rose)
```

```
ggplot(data=predicted.rose, aes(x=rank, y=probability.of.recommended)) +
geom_point(aes(color=is_promoted), alpha=1, shape=4, stroke=2) +
xlab("Index") +
ylab("Predicted probability of recommending for promotion")
```




```
confusion_matrix(logistic_regres)
```

```
##           Predicted 0 Predicted 1 Total
## Actual 0        15562        5040 20602
## Actual 1         4489       16015 20504
## Total           20051       21055 41106
```

```
pdata <- predict(logistic_regres,newdata=test_feat_imp,type="response")
data.rose$is_promoted=as.factor(data.rose$is_promoted)
test_feat_imp$is_promoted=as.factor(test_feat_imp$is_promoted)
pdataF<- as.factor(ifelse(test=as.numeric(pdata>0.54)==0,yes=0,no=1))
```

Confusion Matrix and AUC for Logostic Regression using ROSE

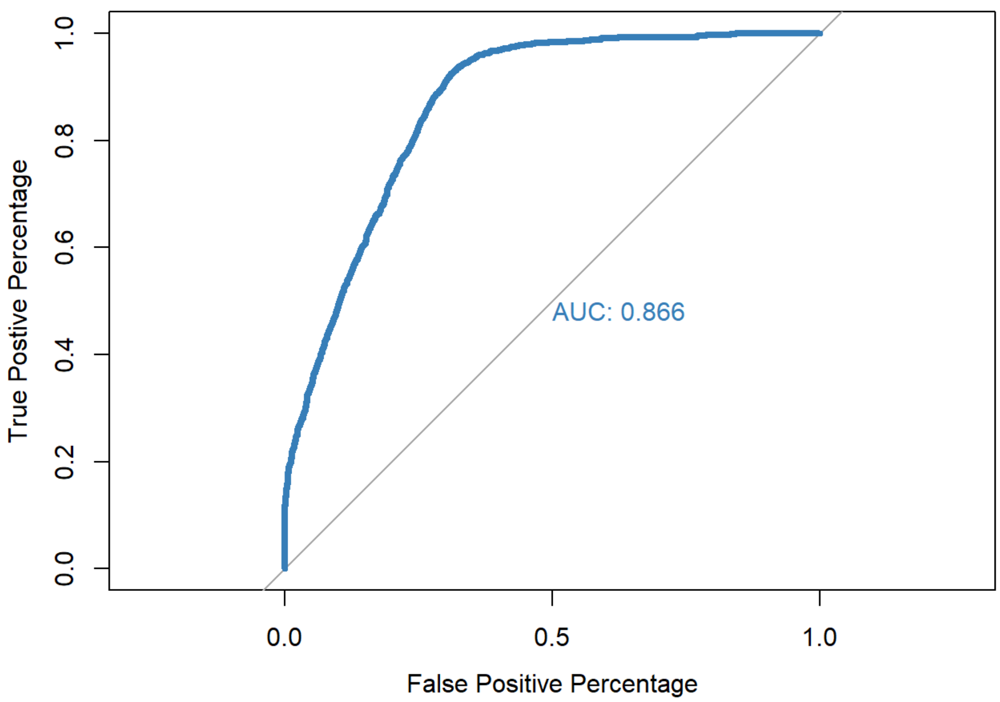
```
confusionMatrix(pdataF,test_feat_imp$is_promoted)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    0    1
##          0 9694  248
##          1 2882  878
##                                           
##                Accuracy : 0.7716          
##                  95% CI : (0.7644, 0.7786)
##     No Information Rate : 0.9178          
##     P-Value [Acc > NIR] : 1               
##                                           
##                   Kappa : 0.2666          
##                                           
##  Mcnemar's Test P-Value : <2e-16          
##                                           
##             Sensitivity : 0.7708          
##             Specificity : 0.7798          
##          Pos Pred Value : 0.9751          
##          Neg Pred Value : 0.2335          
##              Prevalence : 0.9178          
##          Detection Rate : 0.7075          
##    Detection Prevalence : 0.7256          
##       Balanced Accuracy : 0.7753          
##                                           
##        'Positive' Class : 0               
## 
```

```
roc(test_feat_imp$is_promoted,pdata,plot=TRUE, legacy.axes=TRUE, xlab="False Positive Percentage", ylab="True Postive Percentage", col="#377eb8", lwd=4,print.auc= TRUE)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```



```
##
## Call:
## roc.default(response = test_feat_imp$is_promoted, predictor = pdata,      plot = TRUE, legacy.axes = TRUE
, xlab = "False Positive Percentage",      ylab = "True Postive Percentage", col = "#377eb8", lwd = 4,      pr
int.auc = TRUE)
##
## Data: pdata in 12576 controls (test_feat_imp$is_promoted 0) < 1126 cases (test_feat_imp$is_promoted 1).
## Area under the curve: 0.8663
```

Stepwise variable selection using BIC didn't make any change to the accuracy

Stepwise variable selection using AIC didn't make any change to the accuracy

Handling Imbalance using SMOTE

```
balanced.data <- SMOTE(is_promoted ~., train_feat_imp)
table(balanced.data$is_promoted)
```

```
##
##     0     1
## 14168 10626
```

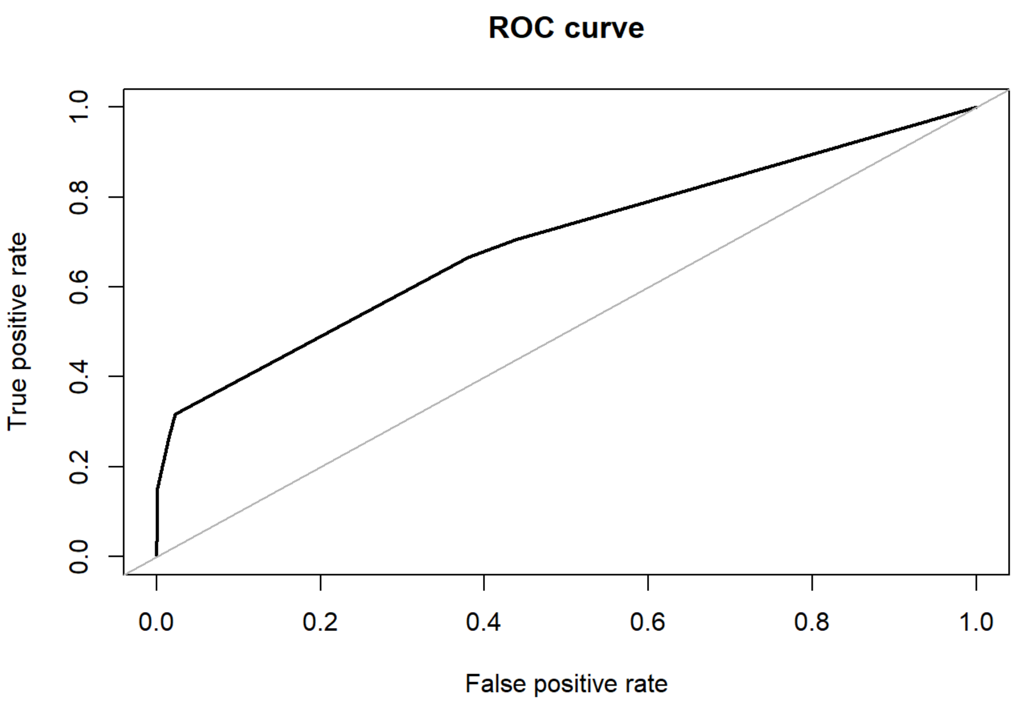
Recursive Partitioning(rpart) with SMOTE data

```
tree.smote<-rpart(is_promoted~.,data=balanced.data)
```

```
pred.tree.smote<-predict(tree.smote,newdata=test_feat_imp,type='prob')
pdataF<- as.factor(ifelse(test=as.numeric(pred.tree.smote[,1]>0.54)==0,no=0,yes=1))
data.frame(pdataF)
```

Confusion Matrix and AUC for Decision tree of SMOTE data

```
confmat.tree.smote<-confusionMatrix(pdataF,test_feat_imp$is_promoted)
roc.curve(test_feat_imp$is_promoted,pred.tree.smote[,2])
```




```
## Area under the curve (AUC): 0.700
```

Random Forest using SMOTE

```
rfsmote<-randomForest(is_promoted ~., data=balanced.data,importance=TRUE)
```

Fine tuning parameters of Random Forest model

```
rfsmotetune1<-randomForest(is_promoted ~., data=balanced.data,ntree=500,mtry=10,importance=TRUE)
```

```
# Predicting on train set
predTrain.smote<-predict(rfrosetune1,balanced.data,type='class')
# Checking classification accuracy
table(predTrain.smote,balanced.data$is_promoted)
```

```
##
## predTrain.smote     0     1
##               0 12137  1336
##               1  2031  9290
```

```
# Predicting on validation set
predValid.smote<-predict(rfsmotetune1,test_feat_imp,type='class')
# Checking classification accuracy
table(predValid.smote,test_feat_imp$is_promoted)
```

```
##
## predValid.smote     0     1
##               0 11112   434
##               1  1464   692
```

Accuracy of the Random forest (SMOTE)

```
mean(predValid.smote==test_feat_imp$is_promoted)
```

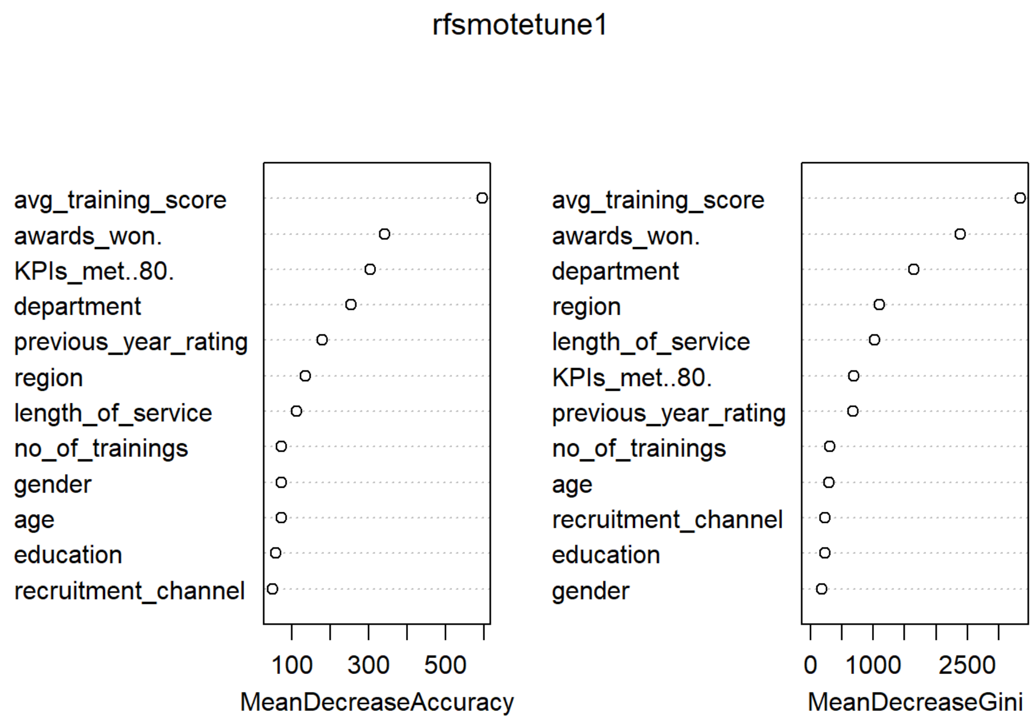
```
## [1] 0.8614801
```

Important Variables obtained from applying random forest

```
importance(rfsmotetune1)
```

```
##                              0           1 MeanDecreaseAccuracy
## department           268.04929 -126.677703            253.77120
## region               148.12493   24.174184            135.54397
## education             62.29687   -8.063388             58.57007
## gender                82.53027    0.903669             73.64229
## recruitment_channel   57.41746   -3.008043             49.78499
## previous_year_rating 186.65951   35.330936            180.34204
## KPIs_met..80.        270.76057  131.416118            304.35751
## awards_won.          331.21523  127.256519            342.72830
## age                   77.53427  -12.869780             73.27413
## length_of_service    109.81578   29.971277            113.16546
## avg_training_score   478.82856   11.391747            595.50795
## no_of_trainings       70.40247   39.811463             74.34137
##                      MeanDecreaseGini
## department                  1644.8899
## region                      1094.8908
## education                    229.3065
## gender                       186.2247
## recruitment_channel          230.0641
## previous_year_rating         683.6114
## KPIs_met..80.                697.1368
## awards_won.                 2390.0559
## age                          297.5732
## length_of_service           1018.8471
## avg_training_score          3333.4610
## no_of_trainings              313.0075
```

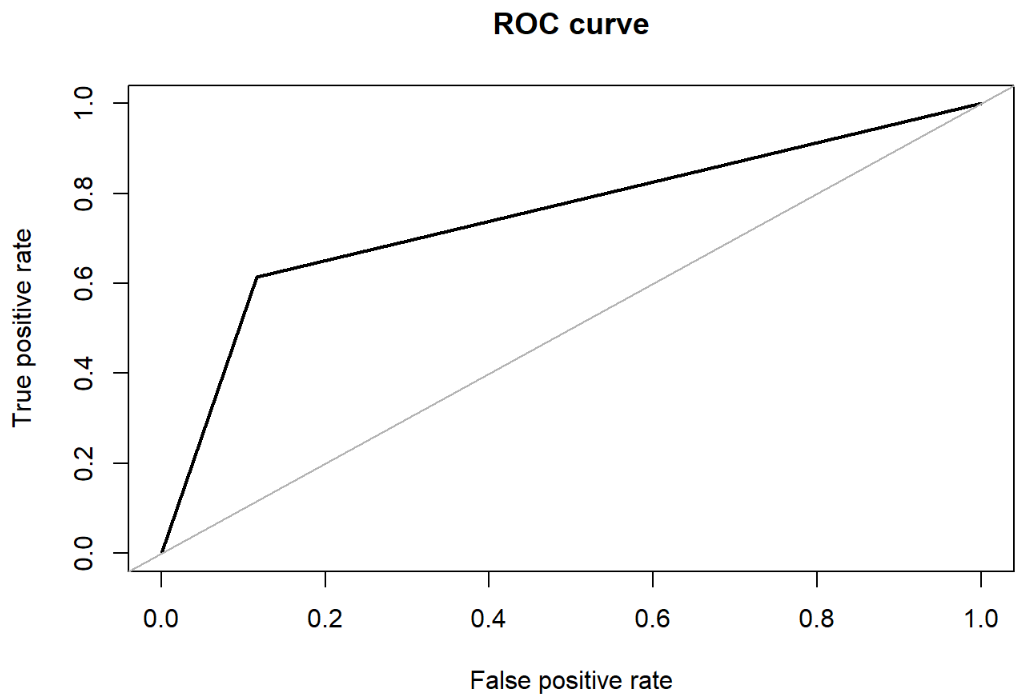
```
varImpPlot(rfsmotetune1)
```



AUC of Random Forest od

SMOTE data

```
roc.curve(test_feat_imp$is_promoted,predValid.smote)
```




```
## Area under the curve (AUC): 0.749
```

Logistic Regression with SMOTE data

```
logistic_smote <- glm( is_promoted ~. ,data=balanced.data,na.action = na.omit,family="binomial")
summary(logistic_smote)
```

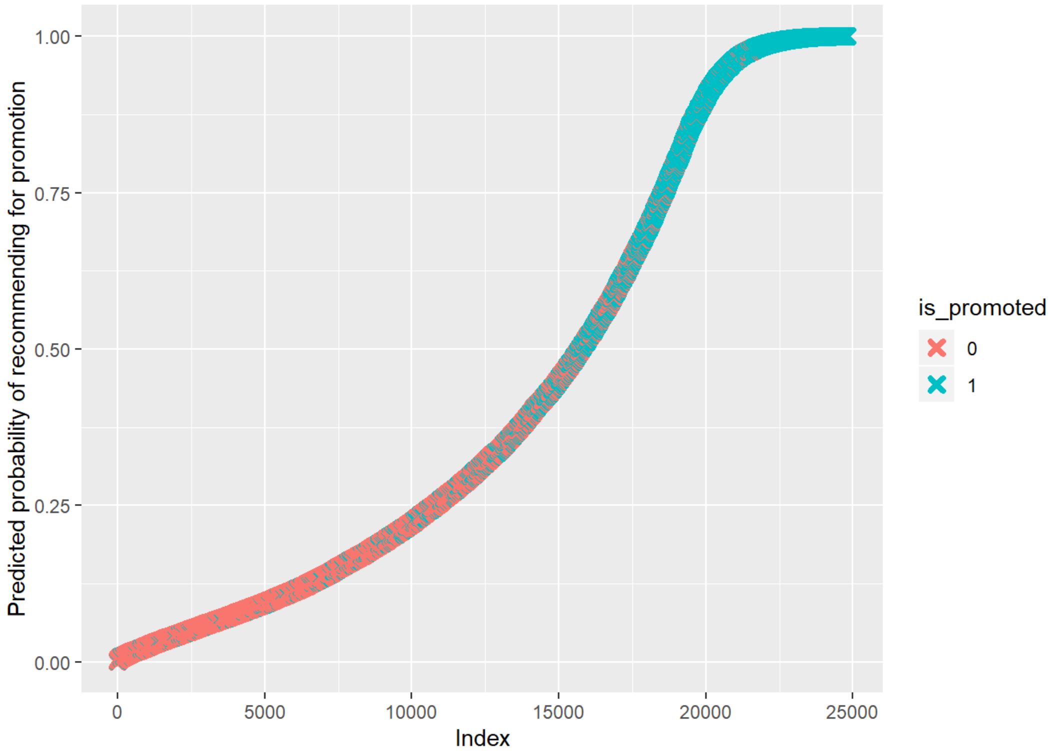
```
##
## Call:
## glm(formula = is_promoted ~ ., family = "binomial", data = balanced.data,
##     na.action = na.omit)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2263  -0.6344  -0.2881   0.3967   3.8303
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -17.871698   0.382050 -46.778  < 2e-16 ***
## departmentFinance             4.039550   0.130225  31.020  < 2e-16 ***
## departmentHR                  5.769005   0.163736  35.234  < 2e-16 ***
## departmentLegal               3.669768   0.186966  19.628  < 2e-16 ***
## departmentOperations          4.286117   0.107787  39.765  < 2e-16 ***
## departmentProcurement         2.579754   0.086942  29.672  < 2e-16 ***
## departmentR&D                -0.566742   0.141300  -4.011 6.05e-05 ***
## departmentSales & Marketing   5.952396   0.137146  43.402  < 2e-16 ***
## departmentTechnology          0.945361   0.071794  13.168  < 2e-16 ***
## regionregion_10              -0.368805   0.247994  -1.487 0.136974
## regionregion_11              -0.587577   0.220247  -2.668 0.007635 **
## regionregion_12              -0.889080   0.275185  -3.231 0.001234 **
## regionregion_13              -0.098531   0.191051  -0.516 0.606043
## regionregion_14              -0.437227   0.236039  -1.852 0.063975 .
## regionregion_15              -0.238695   0.192023  -1.243 0.213849
## regionregion_16              -0.405895   0.211634  -1.918 0.055123 .
## regionregion_17               0.257060   0.213692   1.203 0.228999
## regionregion_18               0.015420   0.886525   0.017 0.986122
## regionregion_19              -0.342460   0.229175  -1.494 0.135093
## regionregion_2               -0.156680   0.178743  -0.877 0.380724
## regionregion_20              -0.269687   0.231047  -1.167 0.243113
## regionregion_21              -1.169564   0.349697  -3.345 0.000824 ***
## regionregion_22               0.063579   0.180729   0.352 0.724996
## regionregion_23               0.212408   0.208352   1.019 0.307980
## regionregion_24              -0.923995   0.320069  -2.887 0.003891 **
## regionregion_25               0.154937   0.218038   0.711 0.477336
## regionregion_26              -0.402769   0.198830  -2.026 0.042795 *
## regionregion_27              -0.392958   0.205398  -1.913 0.055729 .
## regionregion_28               0.241027   0.204469   1.179 0.238480
## regionregion_29              -1.455878   0.254340  -5.724 1.04e-08 ***
## regionregion_3               -0.189728   0.290092  -0.654 0.513094
## regionregion_30              -0.093701   0.236499  -0.396 0.691957
## regionregion_31              -0.605157   0.207918  -2.911 0.003608 **
## regionregion_32              -0.756750   0.244584  -3.094 0.001975 **
## regionregion_33              -0.575389   0.388590  -1.481 0.138684
## regionregion_34              -1.541784   0.423701  -3.639 0.000274 ***
## regionregion_4                0.576254   0.194153   2.968 0.002997 **
## regionregion_5               -0.937778   0.258909  -3.622 0.000292 ***
## regionregion_6               -0.687898   0.257631  -2.670 0.007583 **
## regionregion_7                0.198550   0.182347   1.089 0.276215
## regionregion_8               -0.297703   0.248118  -1.200 0.230201
## regionregion_9               -0.715943   0.342762  -2.089 0.036730 *
## educationBelow Secondary     -0.031491   0.137661  -0.229 0.819058
## educationMaster's & above     0.413470   0.042151   9.809  < 2e-16 ***
## educationUnavailable          0.082292   0.099861   0.824 0.409903
## gender1                       0.395039   0.039019  10.124  < 2e-16 ***
## recruitment_channelreferred   0.104466   0.108554   0.962 0.335878
## recruitment_channelsourcing   0.143827   0.036822   3.906 9.38e-05 ***
## previous_year_rating1        -1.507196   0.124707 -12.086  < 2e-16 ***
## previous_year_rating2        -0.140489   0.100115  -1.403 0.160535
## previous_year_rating3        -0.049113   0.077736  -0.632 0.527522
## previous_year_rating4        -0.223561   0.082983  -2.694 0.007059 **
## previous_year_rating5         0.430842   0.078725   5.473 4.43e-08 ***
## KPIs_met..80.1                1.213696   0.038436  31.577  < 2e-16 ***
## awards_won.1                  3.832121   0.091160  42.037  < 2e-16 ***
## age(40,50]                    0.053772   0.056290   0.955 0.339442
## age(50,60]                   -0.230823   0.096405  -2.394 0.016652 *
## age[20,30]                    0.207174   0.047001   4.408 1.04e-05 ***
## length_of_service            -0.011216   0.005725  -1.959 0.050108 .
## avg_training_score            0.198298   0.003728  53.187  < 2e-16 ***
## no_of_trainings              -0.239283   0.037848  -6.322 2.58e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 33864  on 24793  degrees of freedom
```

```
## Residual deviance: 19181  on 24733  degrees of freedom
## AIC: 19303
##
## Number of Fisher Scoring iterations: 6
```

```
#probablity_pred
predicted.smote<-data.frame(probability.of.recommended=logistic_smote$fitted.values,is_promoted=balanced.dat
a$is_promoted)
predicted.smote <- predicted.smote[order(predicted.smote$probability.of.recommended, decreasing=FALSE),]
predicted.smote$rank <- 1:nrow(predicted.smote)
```

```
ggplot(data=predicted.smote, aes(x=rank, y=probability.of.recommended)) +
geom_point(aes(color=is_promoted), alpha=1, shape=4, stroke=2) +
xlab("Index") +
ylab("Predicted probability of recommending for promotion")
```




```
confusion_matrix(logistic_smote)
```

```
##           Predicted 0 Predicted 1 Total
## Actual 0        12801        1367 14168
## Actual 1         2935        7691 10626
## Total           15736        9058 24794
```

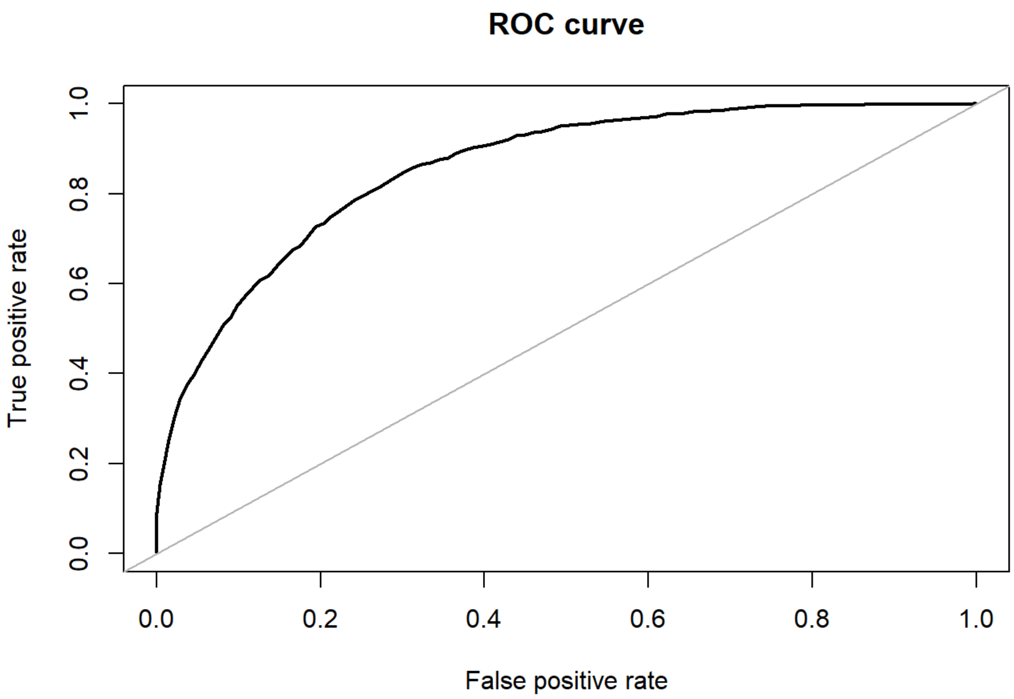
```
psmote <- predict(logistic_smote,newdata=test_feat_imp,type="response")
balanced.data$is_promoted=as.factor(balanced.data$is_promoted)
test_feat_imp$is_promoted=as.factor(test_feat_imp$is_promoted)
smotedataF<- as.factor(ifelse(test=as.numeric(psmote>0.54)==0,yes=0,no=1))
```

Confusion Matrix and AUC of Logistic Regression (SMOTE)

```
confusionMatrix(smotedataF,test_feat_imp$is_promoted)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 11635   574
##          1   941   552
##
##                Accuracy : 0.8894
##                  95% CI : (0.8841, 0.8946)
##     No Information Rate : 0.9178
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3617
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9252
##             Specificity : 0.4902
##          Pos Pred Value : 0.9530
##          Neg Pred Value : 0.3697
##              Prevalence : 0.9178
##          Detection Rate : 0.8491
##    Detection Prevalence : 0.8910
##       Balanced Accuracy : 0.7077
##
##        'Positive' Class : 0
##
```

```
roc.curve(test_feat_imp$is_promoted,psmote)
```




```
## Area under the curve (AUC): 0.857
```