HR\_Employee\_Attrition\_Analysis

Data mining Group 10

Satya Charan Yadati, K Kavita Deepthi and M Mohanraj

2018

Table of Contents

[Introduction: 2](#_Toc521340325)

[Objective of the Analysis: 2](#_Toc521340326)

[Dataset Description: 2](#_Toc521340327)

[Methodology: 3](#_Toc521340328)

[Data Import 3](#_Toc521340329)

[Split the data in Dev & Hold Out sample (70:30) 4](#_Toc521340330)

[Exploratory Data Analysis 5](#_Toc521340331)

[Hypothesis 17](#_Toc521340332)

[Build Neural Network Model (Development sample) 17](#_Toc521340333)

[Build CART Model 32](#_Toc521340334)

[Validate CART Model 34](#_Toc521340335)

[Ensembling Model 35](#_Toc521340336)

[Hypothesis Validation 36](#_Toc521340337)

[Ensembling models have improved accuracy 36](#_Toc521340338)

# Introduction:

###### Attrition in business can mean the reduction in staff and employees in a company through normal means, such as retirement and resignation, the loss of customers or clients to old age or growing out of the company’s target demographic. Changes in management style, company structure, or other aspects of the company might cause employees to leave the company voluntarily, resulting in a higher attrition rate.

###### Another possible cause of attrition is when a company eliminates a job completely. There are different turnover rates across industries, with hospitality and retail having higher rates compared to other industries. But a high turnover rate can be costly. When you think about your investment in recruiting and training employees and only having them stay on for a short period of time, you are not getting back a return on your investment.

###### Customer attrition generally has a negative effect on the company’s profits and growth. This analysis addresses the following issues concerning the attrition of an employee with respect to several paramters. In this analysis, we investigate how the general parameters like Education, Department, Monthly Income, OverTime and others impact the attrition of an employee.

# Objective of the Analysis:

###### The specific objective of this analysis is to predict if an employee is going to resign or not. The final goal is reduction in attrition using the data that is provided.This helps the human resources team with idea and knowledge about the concerning reason which can be eliminated. It also majorily helps in cutting down the cost to the company that is incurred when an employee resigns.

###### Our target variable is “Attrition” which is a dichotomous variable with “Yes/No” values.

# Dataset Description:

###### For this analysis we have the HR Employee Attrition Data file consists of 2940rows/data points and 35 columns/attributes which tend to impact the employee attirition. It is identified that some of the attributes in the data set such as EmployeeCount, EmployeeNumber, Over18 and StandardHours are being same for each employee are not related in this analysis.

###### Therefore, the total attributes now left are 31 some of which are categorical and numeric variables.

###### From the metadata file, we have some of the categorical variables are coded as numeric variables which are ‘dummy variables’, such as the following:

###### Education: 1 ‘Below College’ 2 ‘College’ 3 ‘Bachelor’ 4 ‘Master’ 5 ‘Doctor’

###### EnvironmentSatisfaction: 1 ‘Low’ 2 ‘Medium’ 3 ‘High’ 4 ‘Very High’

###### JobInvolvement: 1 ‘Low’ 2 ‘Medium’ 3 ‘High’ 4 ‘Very High’

###### JobSatisfaction: 1 ‘Low’ 2 ‘Medium’ 3 ‘High’ 4 ‘Very High’

###### PerformanceRating: 1 ‘Low’ 2 ‘Good’ 3 ‘Excellent’ 4 ‘Outstanding’

###### RelationshipSatisfaction: 1 ‘Low’ 2 ‘Medium’ 3 ‘High’ 4 ‘Very High’

###### WorkLifeBalance: 1 ‘Bad’ 2 ‘Good’ 3 ‘Better’ 4 ‘Best’

# Methodology:

###### Analyizing the data set by building Neural Network and CART model

## Data Import

setwd("C:/Users/Kavita/Documents/Great Lakes/Assignments/Data Mining")  
getwd()

## [1] "C:/Users/Kavita/Documents/Great Lakes/Assignments/Data Mining"

emp\_data = read.csv("HR\_Employee\_Attrition\_Data.csv")  
str(emp\_data)

## 'data.frame': 2940 obs. of 35 variables:  
## $ Age : int 41 49 37 33 27 32 59 30 38 36 ...  
## $ Attrition : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 1 1 1 ...  
## $ BusinessTravel : Factor w/ 3 levels "Non-Travel","Travel\_Frequently",..: 3 2 3 2 3 2 3 3 2 3 ...  
## $ DailyRate : int 1102 279 1373 1392 591 1005 1324 1358 216 1299 ...  
## $ Department : Factor w/ 3 levels "Human Resources",..: 3 2 2 2 2 2 2 2 2 2 ...  
## $ DistanceFromHome : int 1 8 2 3 2 2 3 24 23 27 ...  
## $ Education : int 2 1 2 4 1 2 3 1 3 3 ...  
## $ EducationField : Factor w/ 6 levels "Human Resources",..: 2 2 5 2 4 2 4 2 2 4 ...  
## $ EmployeeCount : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ EmployeeNumber : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ EnvironmentSatisfaction : int 2 3 4 4 1 4 3 4 4 3 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 1 2 2 1 2 2 1 2 2 2 ...  
## $ HourlyRate : int 94 61 92 56 40 79 81 67 44 94 ...  
## $ JobInvolvement : int 3 2 2 3 3 3 4 3 2 3 ...  
## $ JobLevel : int 2 2 1 1 1 1 1 1 3 2 ...  
## $ JobRole : Factor w/ 9 levels "Healthcare Representative",..: 8 7 3 7 3 3 3 3 5 1 ...  
## $ JobSatisfaction : int 4 2 3 3 2 4 1 3 3 3 ...  
## $ MaritalStatus : Factor w/ 3 levels "Divorced","Married",..: 3 2 3 2 2 3 2 1 3 2 ...  
## $ MonthlyIncome : int 5993 5130 2090 2909 3468 3068 2670 2693 9526 5237 ...  
## $ MonthlyRate : int 19479 24907 2396 23159 16632 11864 9964 13335 8787 16577 ...  
## $ NumCompaniesWorked : int 8 1 6 1 9 0 4 1 0 6 ...  
## $ Over18 : Factor w/ 1 level "Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ OverTime : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 1 2 1 1 1 ...  
## $ PercentSalaryHike : int 11 23 15 11 12 13 20 22 21 13 ...  
## $ PerformanceRating : int 3 4 3 3 3 3 4 4 4 3 ...  
## $ RelationshipSatisfaction: int 1 4 2 3 4 3 1 2 2 2 ...  
## $ StandardHours : int 80 80 80 80 80 80 80 80 80 80 ...  
## $ StockOptionLevel : int 0 1 0 0 1 0 3 1 0 2 ...  
## $ TotalWorkingYears : int 8 10 7 8 6 8 12 1 10 17 ...  
## $ TrainingTimesLastYear : int 0 3 3 3 3 2 3 2 2 3 ...  
## $ WorkLifeBalance : int 1 3 3 3 3 2 2 3 3 2 ...  
## $ YearsAtCompany : int 6 10 0 8 2 7 1 1 9 7 ...  
## $ YearsInCurrentRole : int 4 7 0 7 2 7 0 0 7 7 ...  
## $ YearsSinceLastPromotion : int 0 1 0 3 2 3 0 0 1 7 ...  
## $ YearsWithCurrManager : int 5 7 0 0 2 6 0 0 8 7 ...

## Split the data in Dev & Hold Out sample (70:30)

###### In order to build the model we have to spilt the data in a dev sample of 70% and Hold out sample of 30% from the overall population.

###### There are numerous approaches to achieve data partitioning.

library(dplyr)

##   
## Attaching package: 'dplyr'

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

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

set.seed(200)  
sample = sample.int(n = nrow(emp\_data), size = floor(.70\*nrow(emp\_data)), replace = F)  
train = emp\_data[sample, ]  
test = emp\_data[-sample, ]

## Exploratory Data Analysis

###### EDA is used to understand the data properties, find patterns of data, suggest modeling strategies and ‘debug’ analyses.

###### After the data spilt into train and test data set with 70:30 ratio, let’s now see look at the format of the fields in the training set

str(train)

## 'data.frame': 2058 obs. of 35 variables:  
## $ Age : int 44 31 32 38 43 25 44 55 55 20 ...  
## $ Attrition : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 2 ...  
## $ BusinessTravel : Factor w/ 3 levels "Non-Travel","Travel\_Frequently",..: 1 2 3 3 3 3 3 3 3 3 ...  
## $ DailyRate : int 489 1327 128 268 782 1372 625 147 836 129 ...  
## $ Department : Factor w/ 3 levels "Human Resources",..: 2 2 2 2 2 3 2 2 2 2 ...  
## $ DistanceFromHome : int 23 3 2 2 6 18 4 20 8 4 ...  
## $ Education : int 3 4 1 5 4 1 3 2 3 3 ...  
## $ EducationField : Factor w/ 6 levels "Human Resources",..: 4 4 6 4 5 2 4 6 4 6 ...  
## $ EmployeeCount : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ EmployeeNumber : int 1570 1716 1733 2030 1960 2464 2088 284 1536 690 ...  
## $ EnvironmentSatisfaction : int 2 2 4 4 2 1 4 2 4 1 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 1 2 ...  
## $ HourlyRate : int 67 73 84 92 50 93 50 37 33 84 ...  
## $ JobInvolvement : int 3 3 2 3 2 4 3 3 3 3 ...  
## $ JobLevel : int 2 3 2 1 4 2 2 2 4 1 ...  
## $ JobRole : Factor w/ 9 levels "Healthcare Representative",..: 3 6 3 7 6 8 1 3 4 3 ...  
## $ JobSatisfaction : int 2 3 1 3 4 3 2 4 3 1 ...  
## $ MaritalStatus : Factor w/ 3 levels "Divorced","Married",..: 2 1 3 2 1 2 3 2 1 3 ...  
## $ MonthlyIncome : int 2042 13675 2176 3057 16627 6232 5933 5415 14756 2973 ...  
## $ MonthlyRate : int 25043 13523 19737 20471 2671 12477 5197 15972 19730 13008 ...  
## $ NumCompaniesWorked : int 4 9 4 6 4 2 9 3 2 1 ...  
## $ Over18 : Factor w/ 1 level "Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ OverTime : Factor w/ 2 levels "No","Yes": 1 1 1 2 2 1 1 2 2 1 ...  
## $ PercentSalaryHike : int 12 12 13 13 14 11 12 19 14 19 ...  
## $ PerformanceRating : int 3 3 3 3 3 3 3 3 3 3 ...  
## $ RelationshipSatisfaction: int 3 1 4 2 3 2 4 4 3 2 ...  
## $ StandardHours : int 80 80 80 80 80 80 80 80 80 80 ...  
## $ StockOptionLevel : int 1 1 0 1 1 0 0 1 3 0 ...  
## $ TotalWorkingYears : int 17 9 9 6 21 6 10 12 21 1 ...  
## $ TrainingTimesLastYear : int 3 3 5 0 3 3 2 4 2 2 ...  
## $ WorkLifeBalance : int 4 3 3 1 2 2 2 3 3 3 ...  
## $ YearsAtCompany : int 3 2 6 1 1 3 5 10 5 1 ...  
## $ YearsInCurrentRole : int 2 2 2 0 0 2 2 7 0 0 ...  
## $ YearsSinceLastPromotion : int 1 2 0 0 0 1 2 0 0 0 ...  
## $ YearsWithCurrManager : int 2 2 4 1 0 2 3 8 2 0 ...

###### Lets look at the proportion of attrition to understand the percentage of employees leaving the organization

table(train$Attrition)

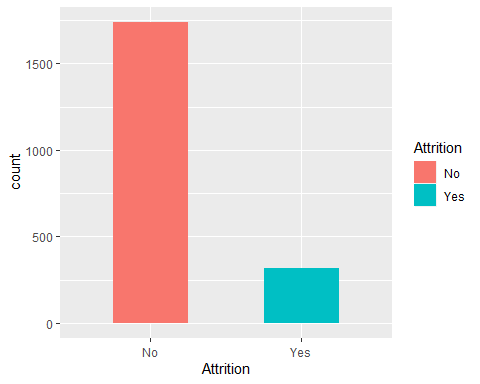
##   
## No Yes   
## 1740 318

prop.table(table(train$Attrition))

##   
## No Yes   
## 0.845481 0.154519

###### We see approximately 16% of employees leaving the organization

library(ggplot2)  
ggplot(train, aes(Attrition, fill=Attrition)) + geom\_bar(aes(fill=Attrition), width = 0.5) +   
 theme(axis.text.x = element\_text(angle=0, vjust=0.6))



#### *Identify columns which are of no use. drop those columns*

###### As mentioned in the dataset description above, we have the structre of emp\_data where some of the attributes such as *EmployeeCount,Over18* and *StandardHours* can be removed from the dataset as these are same of each employee along with *EmployeeNumnber* as it is just an identifier.

train$EmployeeCount = NULL  
train$EmployeeNumber = NULL  
train$Over18 = NULL  
train$StandardHours = NULL  
str(train)

## 'data.frame': 2058 obs. of 31 variables:  
## $ Age : int 44 31 32 38 43 25 44 55 55 20 ...  
## $ Attrition : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 2 ...  
## $ BusinessTravel : Factor w/ 3 levels "Non-Travel","Travel\_Frequently",..: 1 2 3 3 3 3 3 3 3 3 ...  
## $ DailyRate : int 489 1327 128 268 782 1372 625 147 836 129 ...  
## $ Department : Factor w/ 3 levels "Human Resources",..: 2 2 2 2 2 3 2 2 2 2 ...  
## $ DistanceFromHome : int 23 3 2 2 6 18 4 20 8 4 ...  
## $ Education : int 3 4 1 5 4 1 3 2 3 3 ...  
## $ EducationField : Factor w/ 6 levels "Human Resources",..: 4 4 6 4 5 2 4 6 4 6 ...  
## $ EnvironmentSatisfaction : int 2 2 4 4 2 1 4 2 4 1 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 1 2 ...  
## $ HourlyRate : int 67 73 84 92 50 93 50 37 33 84 ...  
## $ JobInvolvement : int 3 3 2 3 2 4 3 3 3 3 ...  
## $ JobLevel : int 2 3 2 1 4 2 2 2 4 1 ...  
## $ JobRole : Factor w/ 9 levels "Healthcare Representative",..: 3 6 3 7 6 8 1 3 4 3 ...  
## $ JobSatisfaction : int 2 3 1 3 4 3 2 4 3 1 ...  
## $ MaritalStatus : Factor w/ 3 levels "Divorced","Married",..: 2 1 3 2 1 2 3 2 1 3 ...  
## $ MonthlyIncome : int 2042 13675 2176 3057 16627 6232 5933 5415 14756 2973 ...  
## $ MonthlyRate : int 25043 13523 19737 20471 2671 12477 5197 15972 19730 13008 ...  
## $ NumCompaniesWorked : int 4 9 4 6 4 2 9 3 2 1 ...  
## $ OverTime : Factor w/ 2 levels "No","Yes": 1 1 1 2 2 1 1 2 2 1 ...  
## $ PercentSalaryHike : int 12 12 13 13 14 11 12 19 14 19 ...  
## $ PerformanceRating : int 3 3 3 3 3 3 3 3 3 3 ...  
## $ RelationshipSatisfaction: int 3 1 4 2 3 2 4 4 3 2 ...  
## $ StockOptionLevel : int 1 1 0 1 1 0 0 1 3 0 ...  
## $ TotalWorkingYears : int 17 9 9 6 21 6 10 12 21 1 ...  
## $ TrainingTimesLastYear : int 3 3 5 0 3 3 2 4 2 2 ...  
## $ WorkLifeBalance : int 4 3 3 1 2 2 2 3 3 3 ...  
## $ YearsAtCompany : int 3 2 6 1 1 3 5 10 5 1 ...  
## $ YearsInCurrentRole : int 2 2 2 0 0 2 2 7 0 0 ...  
## $ YearsSinceLastPromotion : int 1 2 0 0 0 1 2 0 0 0 ...  
## $ YearsWithCurrManager : int 2 2 4 1 0 2 3 8 2 0 ...

#### *Hypothesis and validation*

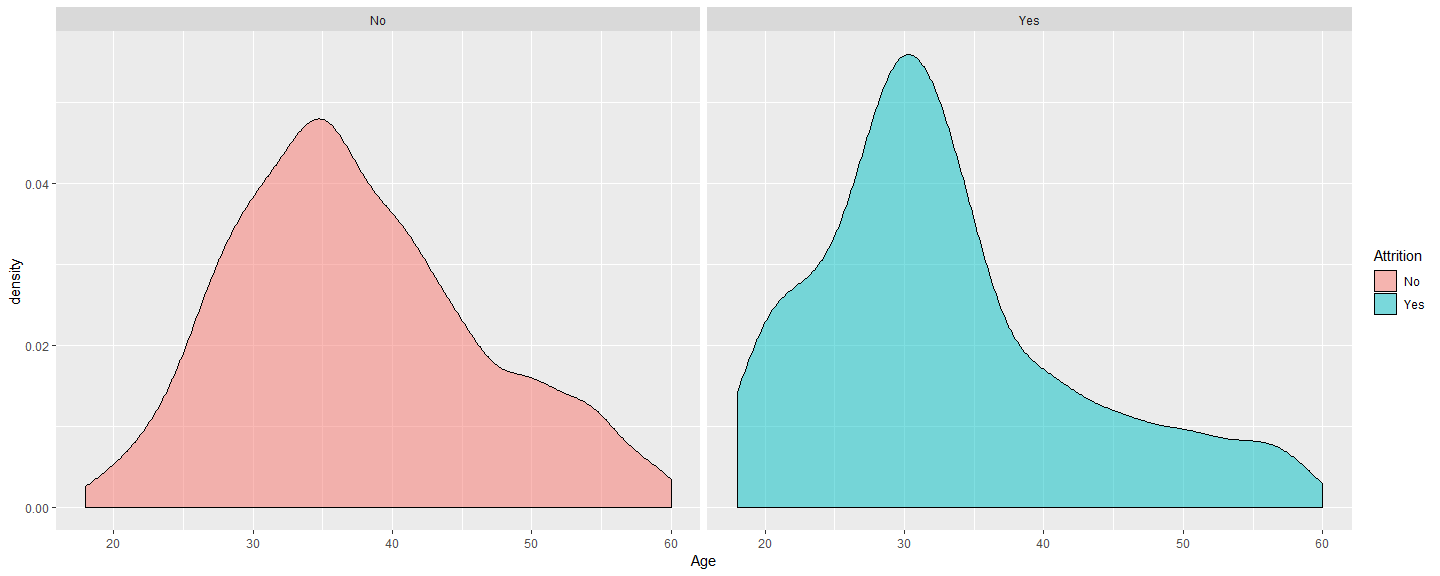
###### In order to write the hypothesis we need to understand the variables that are highly effecting the attrition in the organization. For which, we analyze each influencing variable with attrition.

library(grid)  
library(gridExtra)

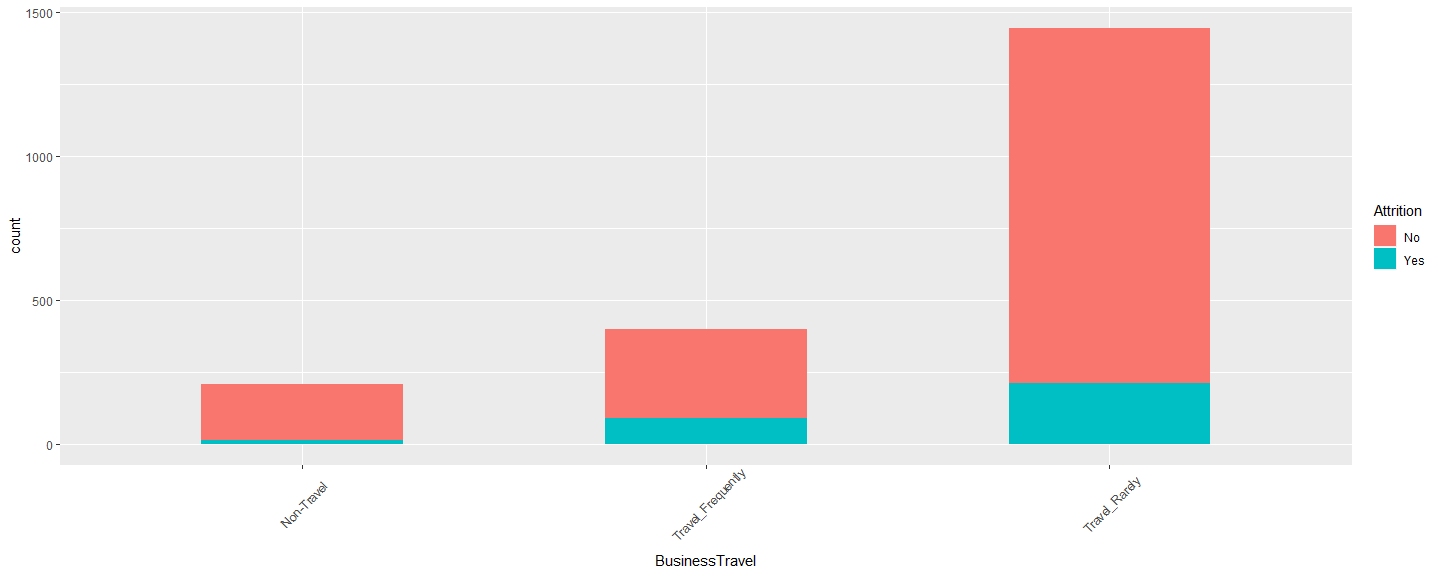
##   
## Attaching package: 'gridExtra'

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

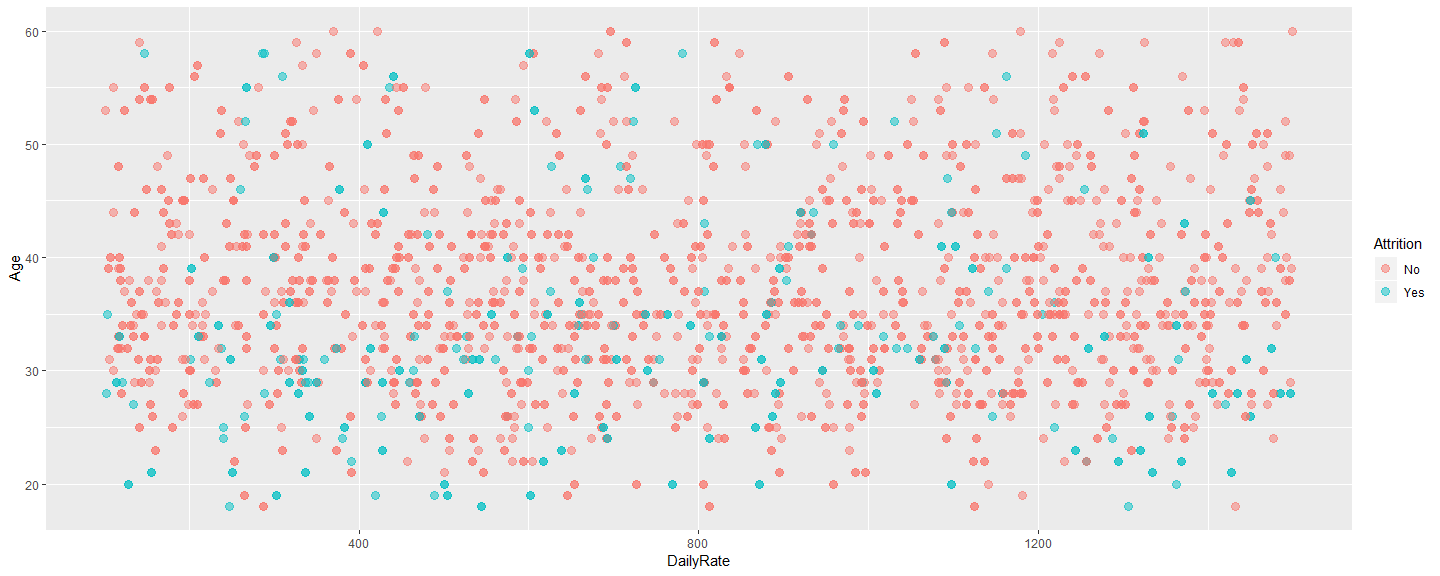
age\_plot = ggplot(train,aes(Age,fill=Attrition))+geom\_density(aes(fill=factor(Attrition)), alpha=0.5)+facet\_grid(~Attrition)  
age\_plot



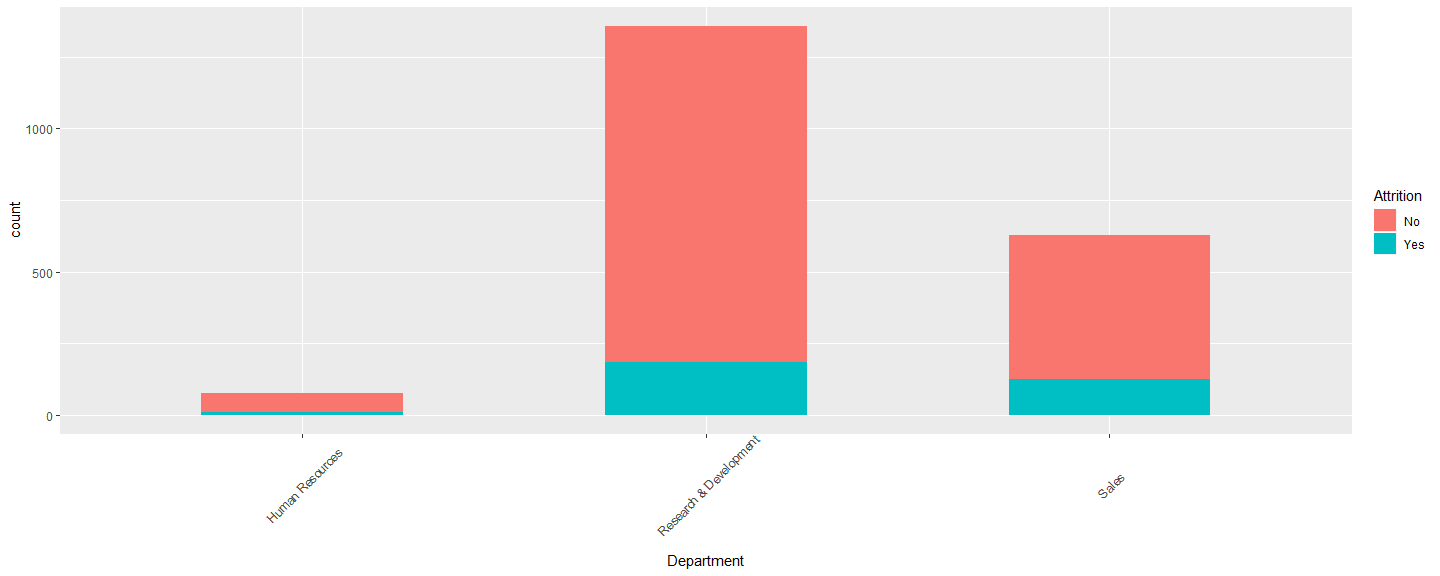
travel\_plot = ggplot(train,aes(BusinessTravel,fill=Attrition))+geom\_bar(width = 0.5)+theme(axis.text.x = element\_text(angle=45, vjust=0.6))  
travel\_plot



rate\_plot = ggplot(train,aes(x=DailyRate, y= Age,color= Attrition))+geom\_point(size=3,alpha = 0.5)  
rate\_plot

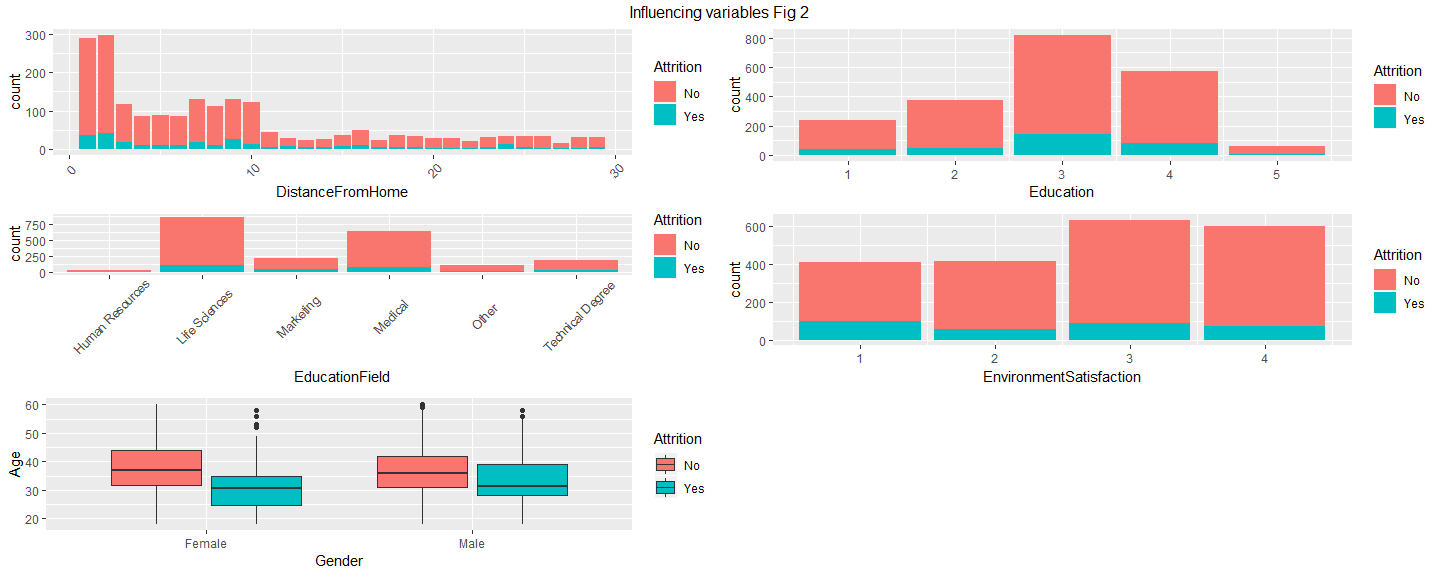


dep\_plot = ggplot(train,aes(Department,fill = Attrition))+geom\_bar(width = 0.5)+theme(axis.text.x = element\_text(angle=45, vjust=0.6))  
dep\_plot



###### From the above analysis, it is identified that employees around 30 years of age, those who travel frequently and R&D dept. are major in attrition and human resources is less in proportion therefore, there no.of attrition is very less.

dist\_plot = ggplot(train,aes(DistanceFromHome,fill=Attrition))+geom\_bar()+theme(axis.text.x = element\_text(angle=45, vjust=0.6))  
edu\_plot = ggplot(train,aes(Education,fill=Attrition))+geom\_bar()  
edufield\_plot = ggplot(train,aes(EducationField,fill=Attrition))+geom\_bar()+theme(axis.text.x = element\_text(angle=45, vjust=0.6))  
env\_plot = ggplot(train,aes(EnvironmentSatisfaction,fill=Attrition))+geom\_bar()  
gen\_plot = ggplot(train,aes(x=Gender,y=Age, fill=Attrition))+geom\_boxplot()  
grid.arrange(dist\_plot,edu\_plot,edufield\_plot,env\_plot,gen\_plot,ncol=2,top = "Influencing variables Fig 2")



Gender\_prop=prop.table(table(train$Gender, train$Attrition))  
Gender\_prop

##   
## No Yes  
## Female 0.34207969 0.05539359  
## Male 0.50340136 0.09912536

Edc\_prop=prop.table(table(train$Education, train$Attrition, train$EducationField))  
Edc\_prop

## , , = Human Resources  
##   
##   
## No Yes  
## 1 0.0004859086 0.0004859086  
## 2 0.0009718173 0.0000000000  
## 3 0.0077745384 0.0019436346  
## 4 0.0014577259 0.0004859086  
## 5 0.0009718173 0.0004859086  
##   
## , , = Life Sciences  
##   
##   
## No Yes  
## 1 0.0413022352 0.0048590865  
## 2 0.0641399417 0.0116618076  
## 3 0.1355685131 0.0247813411  
## 4 0.1078717201 0.0160349854  
## 5 0.0102040816 0.0009718173  
##   
## , , = Marketing  
##   
##   
## No Yes  
## 1 0.0082604470 0.0019436346  
## 2 0.0126336249 0.0034013605  
## 3 0.0310981535 0.0106899903  
## 4 0.0310981535 0.0063168124  
## 5 0.0048590865 0.0000000000  
##   
## , , = Medical  
##   
##   
## No Yes  
## 1 0.0344995141 0.0068027211  
## 2 0.0544217687 0.0082604470  
## 3 0.1088435374 0.0179786200  
## 4 0.0621963071 0.0092322643  
## 5 0.0087463557 0.0000000000  
##   
## , , = Other  
##   
##   
## No Yes  
## 1 0.0024295432 0.0014577259  
## 2 0.0102040816 0.0004859086  
## 3 0.0160349854 0.0014577259  
## 4 0.0179786200 0.0048590865  
## 5 0.0000000000 0.0000000000  
##   
## , , = Technical Degree  
##   
##   
## No Yes  
## 1 0.0087463557 0.0034013605  
## 2 0.0136054422 0.0009718173  
## 3 0.0286686103 0.0111758989  
## 4 0.0194363460 0.0024295432  
## 5 0.0009718173 0.0019436346

##### From the analysis and marginal table, following are the understandings:

###### *Distance From Home*: Contrary to normal assumptions, a mojority of employees who have left the organization are near to the Office.

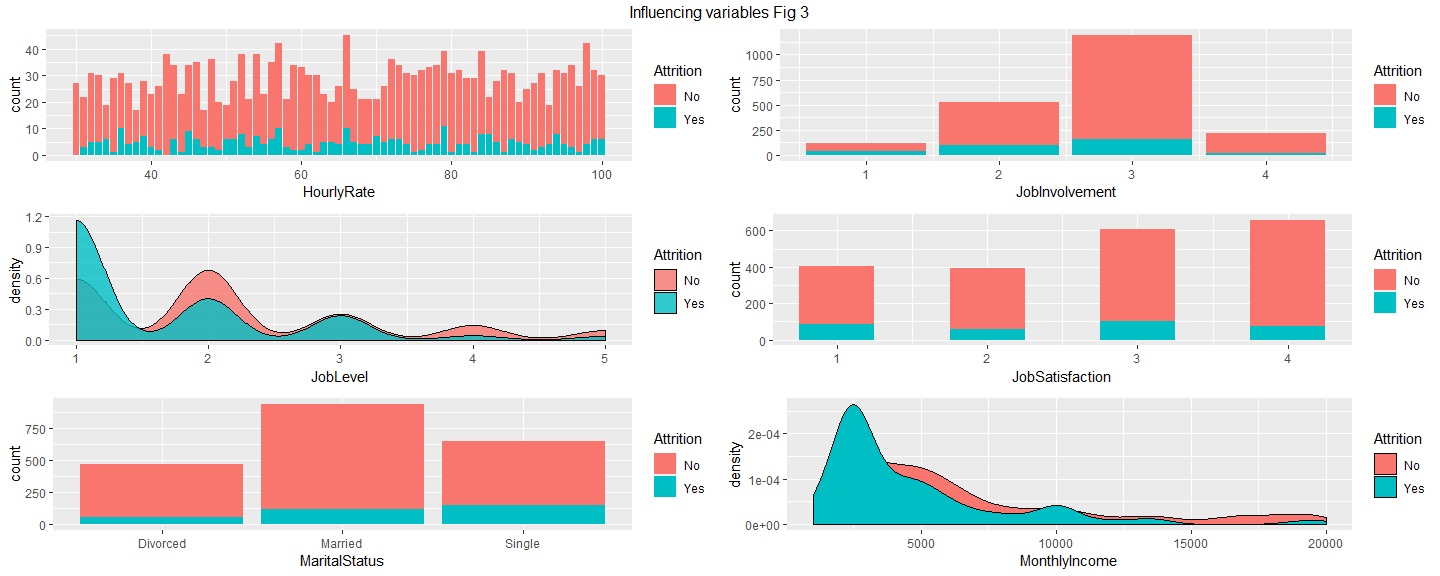
###### *Education*: From the metadata we know that 1 ‘Below College’ 2 ‘College’ 3 ‘Bachelor’ 4 ‘Master’ 5 ‘Doctor’ . Looking at the plot we see that very few Doctors attrite. May be because of less number. And Higher attrition in Bachelors

###### *Education Field*: On lines of the trend in Departments, a minority of HR educated employees leave and it is majorly because of low number of people.

###### *Environment Satisfaction*: Ratings stand for - 1 ‘Low’ 2 ‘Medium’ 3 ‘High’ 4 ‘Very High’ . We don’t see any distinguishable feature.

###### *Gender*: We see that majority of separated employees are Male and the reason might be because around 60.3% of employees in our dataset are Male and they are contributing to major ratio in attrition.

hourly\_plot = ggplot(train,aes(HourlyRate,fill=Attrition))+geom\_bar()  
jobInv\_plot = ggplot(train,aes(JobInvolvement,fill=Attrition))+geom\_bar()  
jobLevel\_plot = ggplot(train,aes(JobLevel,fill=Attrition))+geom\_density(alpha=0.8)  
jobSat\_plot = ggplot(train,aes(JobSatisfaction,fill=Attrition))+geom\_histogram(binwidth = 0.5)  
martial\_plot = ggplot(train,aes(MaritalStatus,fill=Attrition))+geom\_bar()  
mon\_inc\_plot = ggplot(train,aes(MonthlyIncome,fill=Attrition))+geom\_density()  
grid.arrange(hourly\_plot,jobInv\_plot,jobLevel\_plot,jobSat\_plot,martial\_plot,mon\_inc\_plot,ncol=2,top = "Influencing variables Fig 3")



Job\_mon\_prop = prop.table(table(train$JobLevel, train$Attrition, train$MaritalStatus))  
Job\_mon\_prop

## , , = Divorced  
##   
##   
## No Yes  
## 1 0.0631681244 0.0174927114  
## 2 0.0801749271 0.0034013605  
## 3 0.0310981535 0.0029154519  
## 4 0.0174927114 0.0009718173  
## 5 0.0111758989 0.0000000000  
##   
## , , = Married  
##   
##   
## No Yes  
## 1 0.1350826045 0.0296404276  
## 2 0.1525753158 0.0136054422  
## 3 0.0568513120 0.0106899903  
## 4 0.0340136054 0.0004859086  
## 5 0.0218658892 0.0014577259  
##   
## , , = Single  
##   
##   
## No Yes  
## 1 0.0864917396 0.0485908649  
## 2 0.0942662779 0.0160349854  
## 3 0.0335276968 0.0058309038  
## 4 0.0165208941 0.0019436346  
## 5 0.0111758989 0.0014577259

Job\_age\_prop = prop.table(table(train$JobSatisfaction,train$Attrition))  
Job\_age\_prop

##   
## No Yes  
## 1 0.15451895 0.04130224  
## 2 0.16180758 0.02866861  
## 3 0.24538387 0.04859086  
## 4 0.28377065 0.03595724

###### *HourlyRate*: We don’t get much inference from this. There also seems to be no straightforward relation with the Daily Rate of the employees.

###### *Job Involvement*: Ratings stand for 1 ‘Low’ 2 ‘Medium’ 3 ‘High’ 4 ‘Very High’. We see that majority of employees who leave are either Very Highly involved or Low Involved in their Jobs.

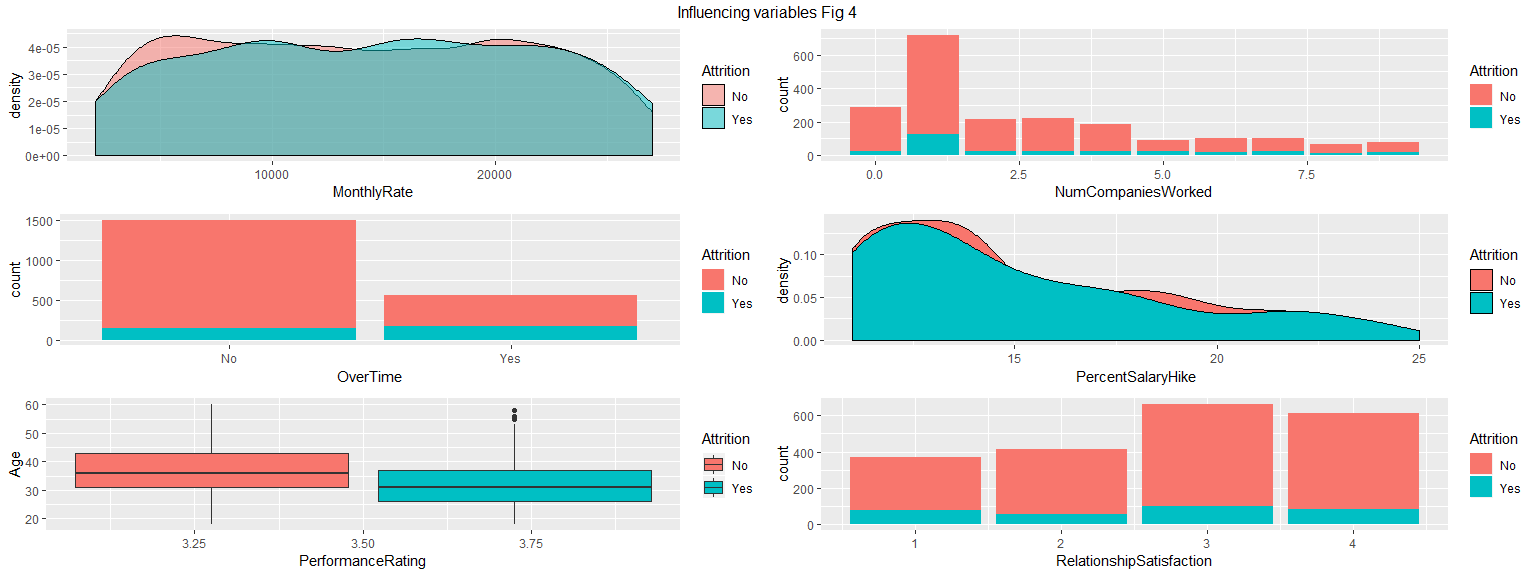
###### *JobLevel*:We have no metadata with regard to the numbers in Job Level. But by looking at proportion of people seems like 1 stands for entry level and 5 stands for highest level in our Dataset. By looking at plot we see that as the Job Level increases the number of people quitting decreases.

###### *Job Satisfaction*: As per metadata 1 ‘Low’ 2 ‘Medium’ 3 ‘High’ 4 ‘Very High’. We see higher attrition levels in among lower Job Satisfaction levels.

###### *Marital Status*: Attrition is on higher side for Single and lowest for Divorced employees. When compared with the job level and marital status, it seems the major attrition is in Job level 1 who are single. This could be because of the higher studies/monetary benefits, considering the facts from Monthly Income variable.

###### *Monthly Income*: We see higher levels of attrition among the lower segment of monthly income. If looked at in isolation, might be due to dissatisfaction of income for the effort out.

monthlyRate\_plot = ggplot(train,aes(MonthlyRate, fill=Attrition))+geom\_density(alpha=0.5)  
numComp\_plot = ggplot(train,aes(NumCompaniesWorked,fill=Attrition))+geom\_bar()  
overTime\_plot = ggplot(train,aes(OverTime,fill=Attrition))+geom\_bar()  
hike\_plot = ggplot(train,aes(PercentSalaryHike, fill=Attrition))+geom\_density()  
perf\_plot = ggplot(train,aes(x=PerformanceRating, y=Age,fill = Attrition))+geom\_boxplot()  
RelSat\_plot = ggplot(train,aes(RelationshipSatisfaction,fill = Attrition))+geom\_bar()  
grid.arrange(monthlyRate\_plot,numComp\_plot,overTime\_plot,hike\_plot,perf\_plot,RelSat\_plot,ncol=2,top = "Influencing variables Fig 4")



###### *Monthly Rate*: We don’t see any inferable trend from this. Also no straightforwad relation with Monthly Income.

###### *Number of Companies Worked*: We see a clear indication that many people who have worked only in One company before quit a lot.

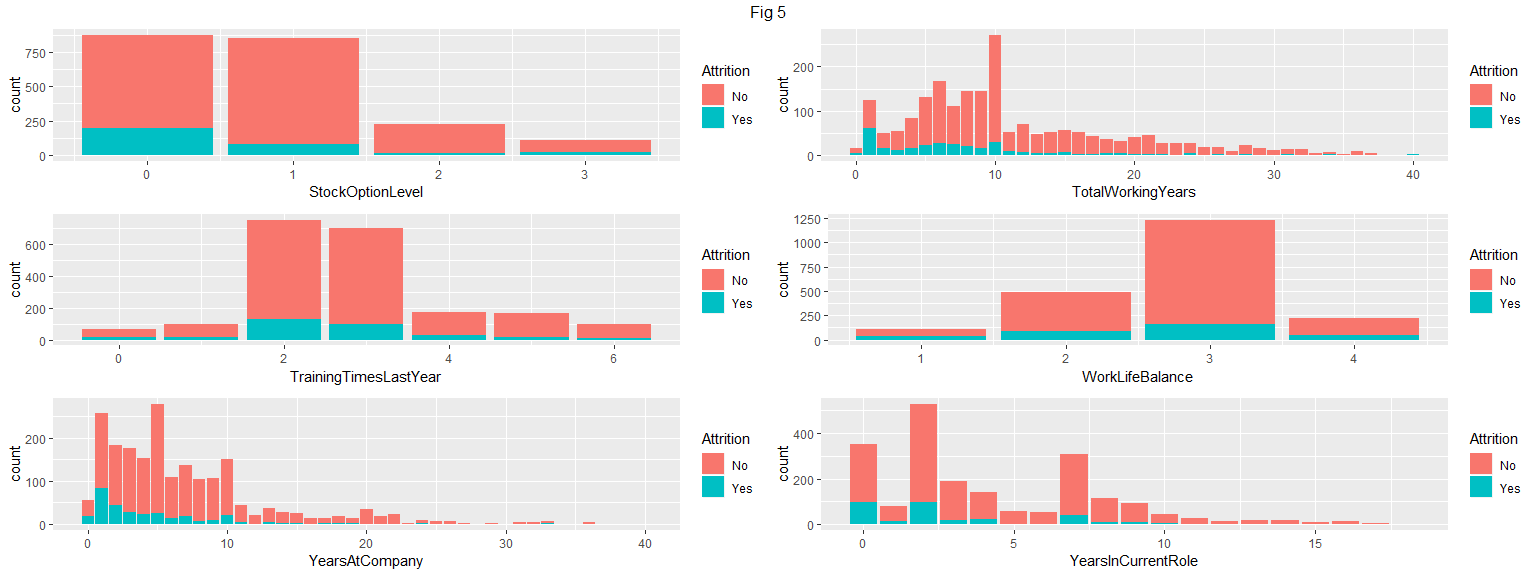
###### *Over Time*: Larger Proportion of Overtime Employees are quitting.

###### *Percent Salary Hike*: We see that people with less than 15% hike have more chances to leave.

###### *Performance Rating*: 1 ‘Low’ 2 ‘Good’ 3 ‘Excellent’ 4 ‘Outstanding’. We see that we have employees of only 3 and 4 ratings. Lesser proportion of 4 raters quit.

###### *Relationship Satisfaction*: 1 ‘Low’ 2 ‘Medium’ 3 ‘High’ 4 ‘Very High’. Higher number of people with 3 or more rating are quitiing. But Larger proportions of 1 & 2 rating are quitting.

StockPlot = ggplot(train,aes(StockOptionLevel,fill = Attrition))+geom\_bar()  
workingYearsPlot = ggplot(train,aes(TotalWorkingYears,fill = Attrition))+geom\_bar()  
TrainTimesPlot = ggplot(train,aes(TrainingTimesLastYear,fill = Attrition))+geom\_bar()  
WLBPlot = ggplot(train,aes(WorkLifeBalance,fill = Attrition))+geom\_bar()  
YearAtComPlot = ggplot(train,aes(YearsAtCompany,fill = Attrition))+geom\_bar()  
YearInCurrPlot = ggplot(train,aes(YearsInCurrentRole,fill = Attrition))+geom\_bar()  
grid.arrange(StockPlot,workingYearsPlot,TrainTimesPlot,WLBPlot,YearAtComPlot, YearInCurrPlot,ncol=2,top = "Fig 5")



###### *Standard Hours*: Same for all and hence not a significant variable for us.

###### *Stock Option Level*: Larger proportions of levels 1 & 2 quit.

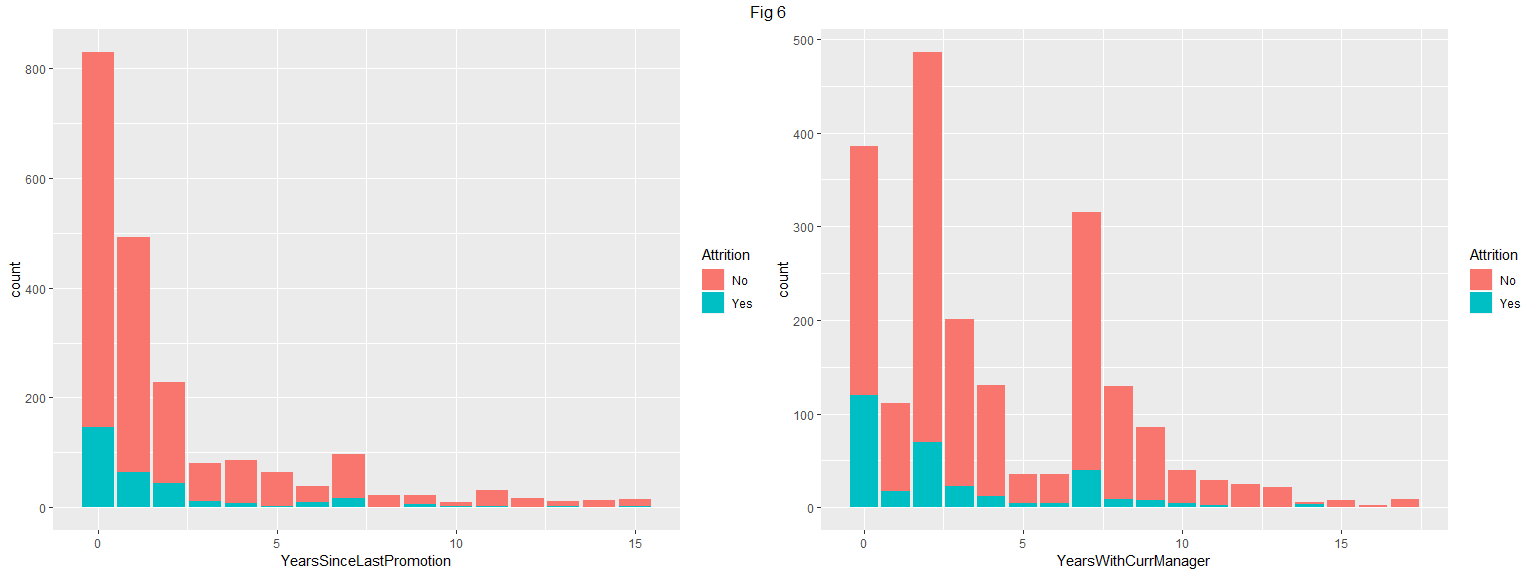
###### *Total Working Years*: We see larger proportions of people with 1 year of experiences quitting the organization also in bracket of 1-10 Years.

###### *Traning Times Last Year*: This indicates the no of training interventions the employee has attended. People who have been trained 2-4 times is an area of concern.

###### *Work Life Balance*:Ratings as per Metadata is 1 ‘Bad’ 2 ‘Good’ 3 ‘Better’ 4 ‘Best’. As expected larger proportion of 1 rating quit, but absolute number wise 2 & 3 are on higher side.

###### *Years at Company*: Larger proportion of new comers are quitting the organization. Which sidelines the recruitment efforts of the organization.

YearsSinceProm = ggplot(train,aes(YearsSinceLastPromotion,fill = Attrition))+geom\_bar()  
YearsCurrManPlot = ggplot(train,aes(YearsWithCurrManager,fill = Attrition))+geom\_bar()  
grid.arrange(YearsSinceProm,YearsCurrManPlot,ncol=2,top = "Fig 6")



###### *Years Since Last Promotion:* Larger proportion of people who have been promoted recently have quit the organization.

###### *Years With Current Manager:* As expected a new Manager is a big cause for quitting.

## Hypothesis

###### Years at Company, Years in Curr Role, Years with Curr Manage, Years Since Last Promotion, Job Level, Monthly Income, Percent Salary Hike,Performance Rating, Overtime,number of companies and Age are the variables that are highly contributing to the Attrition.

###### Among these highly contributed variables, we are considering few variables that have majorily impacted for building the hypothesis.

###### 1. Overtime is the single most variable contributing towards the attrition

###### 2. Employees with less age (young people) with lesser years of experience in the company contributing to the attrition

###### 3. Ensembling different models should improve the accuracy

## Build Neural Network Model (Development sample)

###### Since, we have the training data set we can go ahead and build the NN model with two approaches nnet and neuralnet to see which is giving us the appropriate results.

###### **Note:** Neuralnet package cannot handle categorical inputs. All categorical input variables need to be converted to dummy variables before they can be passed to neuralnet. This is not handled automatically by the package.

##### Fit a Single Hidden Layer Neural Network using Least Squares

library(nnet)  
set.seed(800)  
emp.nn = train  
emp.mod.nn=nnet(Attrition~.,emp.nn,size=18,rang=0.07,Hess=FALSE,decay=15e-4,maxit=1800)

## # weights: 829  
## initial value 1458.216715   
## iter 10 value 874.744602  
## iter 20 value 865.388886  
## iter 30 value 865.379881  
## iter 40 value 863.392420  
## iter 50 value 860.093379  
## iter 60 value 859.551049  
## iter 70 value 859.397944  
## iter 80 value 859.147193  
## iter 90 value 857.193914  
## iter 100 value 854.351000  
## iter 110 value 846.568036  
## iter 120 value 837.767389  
## iter 130 value 806.525525  
## iter 140 value 775.109235  
## iter 150 value 757.923199  
## iter 160 value 747.278186  
## iter 170 value 742.824724  
## iter 180 value 742.529186  
## iter 190 value 739.685891  
## iter 200 value 739.246212  
## iter 210 value 738.123855  
## iter 220 value 729.574293  
## iter 230 value 723.722692  
## iter 240 value 722.861706  
## iter 250 value 720.591760  
## iter 260 value 718.592415  
## iter 270 value 709.943675  
## iter 280 value 705.295869  
## iter 290 value 704.232171  
## iter 300 value 703.080572  
## iter 310 value 702.875009  
## iter 320 value 701.741286  
## iter 330 value 699.853070  
## iter 340 value 693.846037  
## iter 350 value 687.123297  
## iter 360 value 684.602375  
## iter 370 value 682.819149  
## iter 380 value 680.779020  
## iter 390 value 673.866865  
## iter 400 value 669.778329  
## iter 410 value 663.100418  
## iter 420 value 649.623728  
## iter 430 value 635.499178  
## iter 440 value 631.712485  
## iter 450 value 623.129592  
## iter 460 value 612.827112  
## iter 470 value 607.036302  
## iter 480 value 605.348888  
## iter 490 value 605.274941  
## iter 500 value 605.138874  
## iter 510 value 604.856651  
## iter 520 value 604.231414  
## iter 530 value 604.023241  
## iter 540 value 603.994514  
## iter 550 value 603.967914  
## iter 560 value 603.883212  
## iter 570 value 602.904563  
## iter 580 value 601.421737  
## iter 590 value 598.494178  
## iter 600 value 597.314052  
## iter 610 value 596.983877  
## iter 620 value 596.359729  
## iter 630 value 595.990283  
## iter 640 value 595.672871  
## iter 650 value 594.388514  
## iter 660 value 590.904603  
## iter 670 value 582.974546  
## iter 680 value 581.818711  
## iter 690 value 580.873955  
## iter 700 value 579.008748  
## iter 710 value 578.743679  
## iter 720 value 578.523992  
## iter 730 value 578.383239  
## iter 740 value 578.107420  
## iter 750 value 577.018081  
## iter 760 value 576.482543  
## iter 770 value 575.997181  
## iter 780 value 575.606430  
## iter 790 value 575.068289  
## iter 800 value 574.220738  
## iter 810 value 571.394849  
## iter 820 value 569.346991  
## iter 830 value 569.243659  
## iter 840 value 568.899608  
## iter 850 value 568.558398  
## iter 860 value 567.977164  
## iter 870 value 566.969743  
## iter 880 value 566.466976  
## iter 890 value 564.464948  
## iter 900 value 554.646786  
## iter 910 value 543.012728  
## iter 920 value 530.457879  
## iter 930 value 522.866089  
## iter 940 value 519.050763  
## iter 950 value 516.045024  
## iter 960 value 515.982290  
## iter 970 value 515.890011  
## iter 980 value 515.761329  
## iter 990 value 515.604987  
## iter1000 value 515.205154  
## iter1010 value 513.532080  
## iter1020 value 511.959046  
## iter1030 value 510.724385  
## iter1040 value 505.357446  
## iter1050 value 495.191457  
## iter1060 value 485.126313  
## iter1070 value 484.805010  
## iter1080 value 483.006657  
## iter1090 value 482.415180  
## iter1100 value 480.952419  
## iter1110 value 477.005589  
## iter1120 value 468.276646  
## iter1130 value 463.018034  
## iter1140 value 451.679963  
## iter1150 value 439.751728  
## iter1160 value 428.227619  
## iter1170 value 414.566420  
## iter1180 value 413.544918  
## iter1190 value 411.738153  
## iter1200 value 410.870452  
## iter1210 value 408.841215  
## iter1220 value 406.600818  
## iter1230 value 403.991104  
## iter1240 value 397.898435  
## iter1250 value 392.432859  
## iter1260 value 389.328456  
## iter1270 value 386.778233  
## iter1280 value 385.514128  
## iter1290 value 380.423248  
## iter1300 value 375.918481  
## iter1310 value 370.264735  
## iter1320 value 367.431121  
## iter1330 value 366.643079  
## iter1340 value 362.935612  
## iter1350 value 361.544711  
## iter1360 value 361.013717  
## iter1370 value 360.221146  
## iter1380 value 359.531451  
## iter1390 value 359.471844  
## iter1400 value 359.333540  
## iter1410 value 359.159146  
## iter1420 value 358.151571  
## iter1430 value 357.357670  
## iter1440 value 356.916217  
## iter1450 value 354.655341  
## iter1460 value 354.481442  
## iter1470 value 354.336833  
## iter1480 value 353.727128  
## iter1490 value 353.441152  
## iter1500 value 353.168608  
## iter1510 value 353.024001  
## iter1520 value 352.923344  
## iter1530 value 352.796109  
## iter1540 value 352.192576  
## iter1550 value 352.072331  
## iter1560 value 350.487149  
## iter1570 value 349.545404  
## iter1580 value 349.286848  
## iter1590 value 349.001102  
## iter1600 value 348.138993  
## iter1610 value 347.597925  
## iter1620 value 345.808063  
## iter1630 value 345.051061  
## iter1640 value 344.291361  
## iter1650 value 343.511094  
## iter1660 value 342.914294  
## iter1670 value 341.660873  
## iter1680 value 335.319846  
## iter1690 value 334.384105  
## iter1700 value 334.291903  
## iter1710 value 333.616331  
## iter1720 value 332.485200  
## iter1730 value 331.415585  
## iter1740 value 330.059761  
## iter1750 value 328.235960  
## iter1760 value 326.905330  
## iter1770 value 326.209027  
## iter1780 value 325.842705  
## iter1790 value 325.750233  
## iter1800 value 325.744426  
## final value 325.744426   
## stopped after 1800 iterations

emp.mod.nn

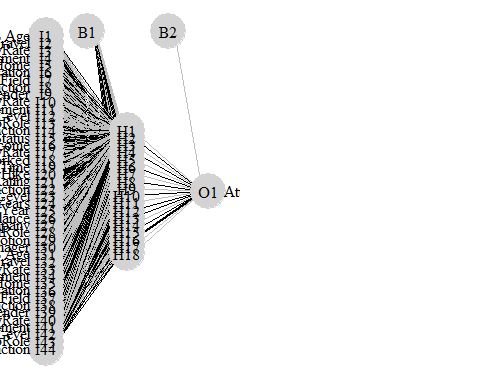
## a 44-18-1 network with 829 weights  
## inputs: Age BusinessTravelTravel\_Frequently BusinessTravelTravel\_Rarely DailyRate DepartmentResearch & Development DepartmentSales DistanceFromHome Education EducationFieldLife Sciences EducationFieldMarketing EducationFieldMedical EducationFieldOther EducationFieldTechnical Degree EnvironmentSatisfaction GenderMale HourlyRate JobInvolvement JobLevel JobRoleHuman Resources JobRoleLaboratory Technician JobRoleManager JobRoleManufacturing Director JobRoleResearch Director JobRoleResearch Scientist JobRoleSales Executive JobRoleSales Representative JobSatisfaction MaritalStatusMarried MaritalStatusSingle MonthlyIncome MonthlyRate NumCompaniesWorked OverTimeYes PercentSalaryHike PerformanceRating RelationshipSatisfaction StockOptionLevel TotalWorkingYears TrainingTimesLastYear WorkLifeBalance YearsAtCompany YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager   
## output(s): Attrition   
## options were - entropy fitting decay=0.0015

#import function from Github  
library(RCurl)

## Loading required package: bitops

require(RCurl)  
   
root.url<-'https://gist.githubusercontent.com/fawda123'  
raw.fun<-paste(  
 root.url,  
 '5086859/raw/cc1544804d5027d82b70e74b83b3941cd2184354/nnet\_plot\_fun.r',  
 sep='/'  
 )  
script<-getURL(raw.fun, ssl.verifypeer = FALSE)  
eval(parse(text = script))  
rm('script','raw.fun')  
  
par(mar=numeric(4),mfrow=c(1,2),family='serif')  
plot(emp.mod.nn)

## Loading required package: scales



###### Prediction table

predict.emp.nn = predict(emp.mod.nn,test,type=("class"))  
table(predict.emp.nn)

## predict.emp.nn  
## No Yes   
## 758 124

##### Confusion Matrix:

actual = test$Attrition  
emp.cm = table(actual, predict.emp.nn)  
emp.cm

## predict.emp.nn  
## actual No Yes  
## No 696 30  
## Yes 62 94

###### A confusion matrix is used to determine the number of true and false positives generated by our predictions. The model generates 696 true negatives (No’s), 94 true positives (Yes’s), while there are 92 false positives.

###### Accuracy of Neural network model

(emp.cm[1]+emp.cm[4])/(nrow(test))

## [1] 0.8956916

###### Neural network gives the accuracy of approximately 90%.

###### Training Neural Network Using BACK PROPOGATION

library(neuralnet)

##   
## Attaching package: 'neuralnet'

## The following object is masked from 'package:dplyr':  
##   
## compute

emp\_data$EmployeeCount = NULL  
emp\_data$EmployeeNumber = NULL  
emp\_data$Over18 = NULL  
emp\_data$StandardHours = NULL  
  
emp\_data\_set = emp\_data  
  
set.seed(200)  
sample\_set = sample.int(n = nrow(emp\_data\_set), size = floor(.70\*nrow(emp\_data\_set)), replace = F)  
train\_set = emp\_data\_set[sample\_set, ]  
test\_set = emp\_data\_set[-sample\_set, ]  
  
str(train\_set)

## 'data.frame': 2058 obs. of 31 variables:  
## $ Age : int 44 31 32 38 43 25 44 55 55 20 ...  
## $ Attrition : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 2 ...  
## $ BusinessTravel : Factor w/ 3 levels "Non-Travel","Travel\_Frequently",..: 1 2 3 3 3 3 3 3 3 3 ...  
## $ DailyRate : int 489 1327 128 268 782 1372 625 147 836 129 ...  
## $ Department : Factor w/ 3 levels "Human Resources",..: 2 2 2 2 2 3 2 2 2 2 ...  
## $ DistanceFromHome : int 23 3 2 2 6 18 4 20 8 4 ...  
## $ Education : int 3 4 1 5 4 1 3 2 3 3 ...  
## $ EducationField : Factor w/ 6 levels "Human Resources",..: 4 4 6 4 5 2 4 6 4 6 ...  
## $ EnvironmentSatisfaction : int 2 2 4 4 2 1 4 2 4 1 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 1 2 ...  
## $ HourlyRate : int 67 73 84 92 50 93 50 37 33 84 ...  
## $ JobInvolvement : int 3 3 2 3 2 4 3 3 3 3 ...  
## $ JobLevel : int 2 3 2 1 4 2 2 2 4 1 ...  
## $ JobRole : Factor w/ 9 levels "Healthcare Representative",..: 3 6 3 7 6 8 1 3 4 3 ...  
## $ JobSatisfaction : int 2 3 1 3 4 3 2 4 3 1 ...  
## $ MaritalStatus : Factor w/ 3 levels "Divorced","Married",..: 2 1 3 2 1 2 3 2 1 3 ...  
## $ MonthlyIncome : int 2042 13675 2176 3057 16627 6232 5933 5415 14756 2973 ...  
## $ MonthlyRate : int 25043 13523 19737 20471 2671 12477 5197 15972 19730 13008 ...  
## $ NumCompaniesWorked : int 4 9 4 6 4 2 9 3 2 1 ...  
## $ OverTime : Factor w/ 2 levels "No","Yes": 1 1 1 2 2 1 1 2 2 1 ...  
## $ PercentSalaryHike : int 12 12 13 13 14 11 12 19 14 19 ...  
## $ PerformanceRating : int 3 3 3 3 3 3 3 3 3 3 ...  
## $ RelationshipSatisfaction: int 3 1 4 2 3 2 4 4 3 2 ...  
## $ StockOptionLevel : int 1 1 0 1 1 0 0 1 3 0 ...  
## $ TotalWorkingYears : int 17 9 9 6 21 6 10 12 21 1 ...  
## $ TrainingTimesLastYear : int 3 3 5 0 3 3 2 4 2 2 ...  
## $ WorkLifeBalance : int 4 3 3 1 2 2 2 3 3 3 ...  
## $ YearsAtCompany : int 3 2 6 1 1 3 5 10 5 1 ...  
## $ YearsInCurrentRole : int 2 2 2 0 0 2 2 7 0 0 ...  
## $ YearsSinceLastPromotion : int 1 2 0 0 0 1 2 0 0 0 ...  
## $ YearsWithCurrManager : int 2 2 4 1 0 2 3 8 2 0 ...

str(test\_set)

## 'data.frame': 882 obs. of 31 variables:  
## $ Age : int 27 38 36 35 29 34 34 34 32 42 ...  
## $ Attrition : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 2 1 ...  
## $ BusinessTravel : Factor w/ 3 levels "Non-Travel","Travel\_Frequently",..: 3 2 3 3 3 3 3 3 2 3 ...  
## $ DailyRate : int 591 216 1299 809 153 1346 419 699 1125 691 ...  
## $ Department : Factor w/ 3 levels "Human Resources",..: 2 2 2 2 2 2 2 2 2 3 ...  
## $ DistanceFromHome : int 2 23 27 16 15 19 7 6 16 8 ...  
## $ Education : int 1 3 3 3 2 2 4 1 1 4 ...  
## $ EducationField : Factor w/ 6 levels "Human Resources",..: 4 2 4 4 2 4 2 4 2 3 ...  
## $ EnvironmentSatisfaction : int 1 4 3 1 4 2 1 2 2 3 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 2 1 2 1 2 ...  
## $ HourlyRate : int 40 44 94 84 49 93 53 83 72 48 ...  
## $ JobInvolvement : int 3 2 3 4 2 3 3 3 1 3 ...  
## $ JobLevel : int 1 3 2 1 2 1 3 1 1 2 ...  
## $ JobRole : Factor w/ 9 levels "Healthcare Representative",..: 3 5 1 3 3 3 6 7 7 8 ...  
## $ JobSatisfaction : int 2 3 3 2 3 4 2 1 1 2 ...  
## $ MaritalStatus : Factor w/ 3 levels "Divorced","Married",..: 2 3 2 2 3 1 3 3 3 2 ...  
## $ MonthlyIncome : int 3468 9526 5237 2426 4193 2661 11994 2960 3919 6825 ...  
## $ MonthlyRate : int 16632 8787 16577 16479 12682 8758 21293 17102 4681 21173 ...  
## $ NumCompaniesWorked : int 9 0 6 0 0 0 0 2 1 0 ...  
## $ OverTime : Factor w/ 2 levels "No","Yes": 1 1 1 1 2 1 1 1 2 1 ...  
## $ PercentSalaryHike : int 12 21 13 13 12 11 11 11 22 11 ...  
## $ PerformanceRating : int 3 4 3 3 3 3 3 3 4 3 ...  
## $ RelationshipSatisfaction: int 4 2 2 3 4 3 3 3 2 4 ...  
## $ StockOptionLevel : int 1 0 2 1 0 1 0 0 0 1 ...  
## $ TotalWorkingYears : int 6 10 17 6 10 3 13 8 10 10 ...  
## $ TrainingTimesLastYear : int 3 2 3 5 3 2 4 2 5 2 ...  
## $ WorkLifeBalance : int 3 3 2 3 3 3 3 3 3 3 ...  
## $ YearsAtCompany : int 2 9 7 5 9 2 12 4 10 9 ...  
## $ YearsInCurrentRole : int 2 7 7 4 5 2 6 2 2 7 ...  
## $ YearsSinceLastPromotion : int 2 1 7 0 0 1 2 1 6 4 ...  
## $ YearsWithCurrManager : int 2 8 7 3 8 2 11 3 7 2 ...

###### It can be observed that all are either integer or factor. Now these factors have to be transformed to numeric.

###### One cannot use directly as.numeric() to convert factors to numeric as it has limitations.

###### First, Lets convert factors having character levels to numeric levels

emp\_data\_transform = emp\_data\_set  
emp\_data\_transform$Attrition=factor(emp\_data\_set$Attrition, levels = c("No","Yes"),labels=c(1,2))  
emp\_data\_transform$BusinessTravel=factor(emp\_data\_set$BusinessTravel, levels = c("Non-Travel","Travel\_Frequently","Travel\_Rarely"),labels=c(1,2,3))  
emp\_data\_transform$Department=factor(emp\_data\_set$Department, levels = c("Human Resources","Research & Development","Sales"),labels=c(1,2,3))  
emp\_data\_transform$EducationField=factor(emp\_data\_set$EducationField, levels = c("Human Resources","Life Sciences","Marketing","Medical","Other","Technical Degree"),labels=c(1,2,3,4,5,6))  
emp\_data\_transform$Gender=factor(emp\_data\_set$Gender, levels = c("Female","Male"),labels=c(1,2))  
emp\_data\_transform$JobRole=factor(emp\_data\_set$JobRole, levels = c("Healthcare Representative","Human Resources","Laboratory Technician","Manager","Manufacturing Director","Research Director","Research Scientist","Sales Executive","Sales Representative"),labels=c(1,2,3,4,5,6,7,8,9))  
emp\_data\_transform$MaritalStatus=factor(emp\_data\_set$MaritalStatus, levels = c("Divorced","Married","Single"),labels=c(1,2,3))  
emp\_data\_transform$OverTime=factor(emp\_data\_set$OverTime, levels = c("No","Yes"),labels=c(1,2))  
str(emp\_data\_transform)

## 'data.frame': 2940 obs. of 31 variables:  
## $ Age : int 41 49 37 33 27 32 59 30 38 36 ...  
## $ Attrition : Factor w/ 2 levels "1","2": 2 1 2 1 1 1 1 1 1 1 ...  
## $ BusinessTravel : Factor w/ 3 levels "1","2","3": 3 2 3 2 3 2 3 3 2 3 ...  
## $ DailyRate : int 1102 279 1373 1392 591 1005 1324 1358 216 1299 ...  
## $ Department : Factor w/ 3 levels "1","2","3": 3 2 2 2 2 2 2 2 2 2 ...  
## $ DistanceFromHome : int 1 8 2 3 2 2 3 24 23 27 ...  
## $ Education : int 2 1 2 4 1 2 3 1 3 3 ...  
## $ EducationField : Factor w/ 6 levels "1","2","3","4",..: 2 2 5 2 4 2 4 2 2 4 ...  
## $ EnvironmentSatisfaction : int 2 3 4 4 1 4 3 4 4 3 ...  
## $ Gender : Factor w/ 2 levels "1","2": 1 2 2 1 2 2 1 2 2 2 ...  
## $ HourlyRate : int 94 61 92 56 40 79 81 67 44 94 ...  
## $ JobInvolvement : int 3 2 2 3 3 3 4 3 2 3 ...  
## $ JobLevel : int 2 2 1 1 1 1 1 1 3 2 ...  
## $ JobRole : Factor w/ 9 levels "1","2","3","4",..: 8 7 3 7 3 3 3 3 5 1 ...  
## $ JobSatisfaction : int 4 2 3 3 2 4 1 3 3 3 ...  
## $ MaritalStatus : Factor w/ 3 levels "1","2","3": 3 2 3 2 2 3 2 1 3 2 ...  
## $ MonthlyIncome : int 5993 5130 2090 2909 3468 3068 2670 2693 9526 5237 ...  
## $ MonthlyRate : int 19479 24907 2396 23159 16632 11864 9964 13335 8787 16577 ...  
## $ NumCompaniesWorked : int 8 1 6 1 9 0 4 1 0 6 ...  
## $ OverTime : Factor w/ 2 levels "1","2": 2 1 2 2 1 1 2 1 1 1 ...  
## $ PercentSalaryHike : int 11 23 15 11 12 13 20 22 21 13 ...  
## $ PerformanceRating : int 3 4 3 3 3 3 4 4 4 3 ...  
## $ RelationshipSatisfaction: int 1 4 2 3 4 3 1 2 2 2 ...  
## $ StockOptionLevel : int 0 1 0 0 1 0 3 1 0 2 ...  
## $ TotalWorkingYears : int 8 10 7 8 6 8 12 1 10 17 ...  
## $ TrainingTimesLastYear : int 0 3 3 3 3 2 3 2 2 3 ...  
## $ WorkLifeBalance : int 1 3 3 3 3 2 2 3 3 2 ...  
## $ YearsAtCompany : int 6 10 0 8 2 7 1 1 9 7 ...  
## $ YearsInCurrentRole : int 4 7 0 7 2 7 0 0 7 7 ...  
## $ YearsSinceLastPromotion : int 0 1 0 3 2 3 0 0 1 7 ...  
## $ YearsWithCurrManager : int 5 7 0 0 2 6 0 0 8 7 ...

###### Now convert these numerical factors into numeric

emp\_data\_transform$Age=as.numeric(as.character(emp\_data\_transform$Age))  
emp\_data\_transform$Attrition=as.numeric(as.character(emp\_data\_transform$Attrition))  
emp\_data\_transform$BusinessTravel=as.numeric(as.character(emp\_data\_transform$BusinessTravel))  
emp\_data\_transform$DailyRate=as.numeric(as.character(emp\_data\_transform$DailyRate))  
emp\_data\_transform$Department=as.numeric(as.character(emp\_data\_transform$Department))  
emp\_data\_transform$DistanceFromHome=as.numeric(as.character(emp\_data\_transform$DistanceFromHome))  
emp\_data\_transform$Education=as.numeric(as.character(emp\_data\_transform$Education))  
emp\_data\_transform$EducationField=as.numeric(as.character(emp\_data\_transform$EducationField))  
emp\_data\_transform$EnvironmentSatisfaction=as.numeric(as.character(emp\_data\_transform$EnvironmentSatisfaction))  
emp\_data\_transform$Gender=as.numeric(as.character(emp\_data\_transform$Gender))  
emp\_data\_transform$HourlyRate=as.numeric(as.character(emp\_data\_transform$HourlyRate))  
emp\_data\_transform$JobInvolvement=as.numeric(as.character(emp\_data\_transform$JobInvolvement))  
emp\_data\_transform$JobLevel=as.numeric(as.character(emp\_data\_transform$JobLevel))  
emp\_data\_transform$JobRole=as.numeric(as.character(emp\_data\_transform$JobRole))  
emp\_data\_transform$JobSatisfaction=as.numeric(as.character(emp\_data\_transform$JobSatisfaction))  
emp\_data\_transform$MaritalStatus=as.numeric(as.character(emp\_data\_transform$MaritalStatus))  
emp\_data\_transform$MonthlyIncome=as.numeric(as.character(emp\_data\_transform$MonthlyIncome))  
emp\_data\_transform$MonthlyRate=as.numeric(as.character(emp\_data\_transform$MonthlyRate))  
emp\_data\_transform$NumCompaniesWorked=as.numeric(as.character(emp\_data\_transform$NumCompaniesWorked))  
emp\_data\_transform$OverTime=as.numeric(as.character(emp\_data\_transform$OverTime))  
emp\_data\_transform$PercentSalaryHike=as.numeric(as.character(emp\_data\_transform$PercentSalaryHike))  
emp\_data\_transform$PerformanceRating=as.numeric(as.character(emp\_data\_transform$PerformanceRating))  
emp\_data\_transform$RelationshipSatisfaction=as.numeric(as.character(emp\_data\_transform$RelationshipSatisfaction))  
emp\_data\_transform$StockOptionLevel=as.numeric(as.character(emp\_data\_transform$StockOptionLevel))  
emp\_data\_transform$TotalWorkingYears=as.numeric(as.character(emp\_data\_transform$TotalWorkingYears))  
emp\_data\_transform$TrainingTimesLastYear=as.numeric(as.character(emp\_data\_transform$TrainingTimesLastYear))  
emp\_data\_transform$WorkLifeBalance=as.numeric(as.character(emp\_data\_transform$WorkLifeBalance))  
emp\_data\_transform$YearsAtCompany=as.numeric(as.character(emp\_data\_transform$YearsAtCompany))  
emp\_data\_transform$YearsInCurrentRole=as.numeric(as.character(emp\_data\_transform$YearsInCurrentRole))  
emp\_data\_transform$YearsSinceLastPromotion=as.numeric(as.character(emp\_data\_transform$YearsSinceLastPromotion))  
emp\_data\_transform$YearsWithCurrManager=as.numeric(as.character(emp\_data\_transform$YearsWithCurrManager))  
str(emp\_data\_transform)

## 'data.frame': 2940 obs. of 31 variables:  
## $ Age : num 41 49 37 33 27 32 59 30 38 36 ...  
## $ Attrition : num 2 1 2 1 1 1 1 1 1 1 ...  
## $ BusinessTravel : num 3 2 3 2 3 2 3 3 2 3 ...  
## $ DailyRate : num 1102 279 1373 1392 591 ...  
## $ Department : num 3 2 2 2 2 2 2 2 2 2 ...  
## $ DistanceFromHome : num 1 8 2 3 2 2 3 24 23 27 ...  
## $ Education : num 2 1 2 4 1 2 3 1 3 3 ...  
## $ EducationField : num 2 2 5 2 4 2 4 2 2 4 ...  
## $ EnvironmentSatisfaction : num 2 3 4 4 1 4 3 4 4 3 ...  
## $ Gender : num 1 2 2 1 2 2 1 2 2 2 ...  
## $ HourlyRate : num 94 61 92 56 40 79 81 67 44 94 ...  
## $ JobInvolvement : num 3 2 2 3 3 3 4 3 2 3 ...  
## $ JobLevel : num 2 2 1 1 1 1 1 1 3 2 ...  
## $ JobRole : num 8 7 3 7 3 3 3 3 5 1 ...  
## $ JobSatisfaction : num 4 2 3 3 2 4 1 3 3 3 ...  
## $ MaritalStatus : num 3 2 3 2 2 3 2 1 3 2 ...  
## $ MonthlyIncome : num 5993 5130 2090 2909 3468 ...  
## $ MonthlyRate : num 19479 24907 2396 23159 16632 ...  
## $ NumCompaniesWorked : num 8 1 6 1 9 0 4 1 0 6 ...  
## $ OverTime : num 2 1 2 2 1 1 2 1 1 1 ...  
## $ PercentSalaryHike : num 11 23 15 11 12 13 20 22 21 13 ...  
## $ PerformanceRating : num 3 4 3 3 3 3 4 4 4 3 ...  
## $ RelationshipSatisfaction: num 1 4 2 3 4 3 1 2 2 2 ...  
## $ StockOptionLevel : num 0 1 0 0 1 0 3 1 0 2 ...  
## $ TotalWorkingYears : num 8 10 7 8 6 8 12 1 10 17 ...  
## $ TrainingTimesLastYear : num 0 3 3 3 3 2 3 2 2 3 ...  
## $ WorkLifeBalance : num 1 3 3 3 3 2 2 3 3 2 ...  
## $ YearsAtCompany : num 6 10 0 8 2 7 1 1 9 7 ...  
## $ YearsInCurrentRole : num 4 7 0 7 2 7 0 0 7 7 ...  
## $ YearsSinceLastPromotion : num 0 1 0 3 2 3 0 0 1 7 ...  
## $ YearsWithCurrManager : num 5 7 0 0 2 6 0 0 8 7 ...

###### Now all the variables are wither intergers or numeric

###### Now we shall partition the data into train and test data

set.seed(200)  
sample\_transform = sample.int(n = nrow(emp\_data\_transform), size = floor(.70\*nrow(emp\_data\_transform)), replace = F)  
trainnew = emp\_data\_transform[sample\_transform, ]  
testnew = emp\_data\_transform[-sample\_transform, ]  
  
str(trainnew)

## 'data.frame': 2058 obs. of 31 variables:  
## $ Age : num 44 31 32 38 43 25 44 55 55 20 ...  
## $ Attrition : num 1 1 1 1 1 1 1 1 1 2 ...  
## $ BusinessTravel : num 1 2 3 3 3 3 3 3 3 3 ...  
## $ DailyRate : num 489 1327 128 268 782 ...  
## $ Department : num 2 2 2 2 2 3 2 2 2 2 ...  
## $ DistanceFromHome : num 23 3 2 2 6 18 4 20 8 4 ...  
## $ Education : num 3 4 1 5 4 1 3 2 3 3 ...  
## $ EducationField : num 4 4 6 4 5 2 4 6 4 6 ...  
## $ EnvironmentSatisfaction : num 2 2 4 4 2 1 4 2 4 1 ...  
## $ Gender : num 2 2 2 2 2 2 2 2 1 2 ...  
## $ HourlyRate : num 67 73 84 92 50 93 50 37 33 84 ...  
## $ JobInvolvement : num 3 3 2 3 2 4 3 3 3 3 ...  
## $ JobLevel : num 2 3 2 1 4 2 2 2 4 1 ...  
## $ JobRole : num 3 6 3 7 6 8 1 3 4 3 ...  
## $ JobSatisfaction : num 2 3 1 3 4 3 2 4 3 1 ...  
## $ MaritalStatus : num 2 1 3 2 1 2 3 2 1 3 ...  
## $ MonthlyIncome : num 2042 13675 2176 3057 16627 ...  
## $ MonthlyRate : num 25043 13523 19737 20471 2671 ...  
## $ NumCompaniesWorked : num 4 9 4 6 4 2 9 3 2 1 ...  
## $ OverTime : num 1 1 1 2 2 1 1 2 2 1 ...  
## $ PercentSalaryHike : num 12 12 13 13 14 11 12 19 14 19 ...  
## $ PerformanceRating : num 3 3 3 3 3 3 3 3 3 3 ...  
## $ RelationshipSatisfaction: num 3 1 4 2 3 2 4 4 3 2 ...  
## $ StockOptionLevel : num 1 1 0 1 1 0 0 1 3 0 ...  
## $ TotalWorkingYears : num 17 9 9 6 21 6 10 12 21 1 ...  
## $ TrainingTimesLastYear : num 3 3 5 0 3 3 2 4 2 2 ...  
## $ WorkLifeBalance : num 4 3 3 1 2 2 2 3 3 3 ...  
## $ YearsAtCompany : num 3 2 6 1 1 3 5 10 5 1 ...  
## $ YearsInCurrentRole : num 2 2 2 0 0 2 2 7 0 0 ...  
## $ YearsSinceLastPromotion : num 1 2 0 0 0 1 2 0 0 0 ...  
## $ YearsWithCurrManager : num 2 2 4 1 0 2 3 8 2 0 ...

str(testnew)

## 'data.frame': 882 obs. of 31 variables:  
## $ Age : num 27 38 36 35 29 34 34 34 32 42 ...  
## $ Attrition : num 1 1 1 1 1 1 1 2 2 1 ...  
## $ BusinessTravel : num 3 2 3 3 3 3 3 3 2 3 ...  
## $ DailyRate : num 591 216 1299 809 153 ...  
## $ Department : num 2 2 2 2 2 2 2 2 2 3 ...  
## $ DistanceFromHome : num 2 23 27 16 15 19 7 6 16 8 ...  
## $ Education : num 1 3 3 3 2 2 4 1 1 4 ...  
## $ EducationField : num 4 2 4 4 2 4 2 4 2 3 ...  
## $ EnvironmentSatisfaction : num 1 4 3 1 4 2 1 2 2 3 ...  
## $ Gender : num 2 2 2 2 1 2 1 2 1 2 ...  
## $ HourlyRate : num 40 44 94 84 49 93 53 83 72 48 ...  
## $ JobInvolvement : num 3 2 3 4 2 3 3 3 1 3 ...  
## $ JobLevel : num 1 3 2 1 2 1 3 1 1 2 ...  
## $ JobRole : num 3 5 1 3 3 3 6 7 7 8 ...  
## $ JobSatisfaction : num 2 3 3 2 3 4 2 1 1 2 ...  
## $ MaritalStatus : num 2 3 2 2 3 1 3 3 3 2 ...  
## $ MonthlyIncome : num 3468 9526 5237 2426 4193 ...  
## $ MonthlyRate : num 16632 8787 16577 16479 12682 ...  
## $ NumCompaniesWorked : num 9 0 6 0 0 0 0 2 1 0 ...  
## $ OverTime : num 1 1 1 1 2 1 1 1 2 1 ...  
## $ PercentSalaryHike : num 12 21 13 13 12 11 11 11 22 11 ...  
## $ PerformanceRating : num 3 4 3 3 3 3 3 3 4 3 ...  
## $ RelationshipSatisfaction: num 4 2 2 3 4 3 3 3 2 4 ...  
## $ StockOptionLevel : num 1 0 2 1 0 1 0 0 0 1 ...  
## $ TotalWorkingYears : num 6 10 17 6 10 3 13 8 10 10 ...  
## $ TrainingTimesLastYear : num 3 2 3 5 3 2 4 2 5 2 ...  
## $ WorkLifeBalance : num 3 3 2 3 3 3 3 3 3 3 ...  
## $ YearsAtCompany : num 2 9 7 5 9 2 12 4 10 9 ...  
## $ YearsInCurrentRole : num 2 7 7 4 5 2 6 2 2 7 ...  
## $ YearsSinceLastPromotion : num 2 1 7 0 0 1 2 1 6 4 ...  
## $ YearsWithCurrManager : num 2 8 7 3 8 2 11 3 7 2 ...

library(neuralnet)  
trainnew.nnbp = neuralnet(formula=Attrition~Age+BusinessTravel+DailyRate+Department+DistanceFromHome+Education+EducationField+EnvironmentSatisfaction+Gender+HourlyRate+JobInvolvement+JobLevel+JobRole+JobSatisfaction+MaritalStatus+MonthlyIncome+MonthlyRate+NumCompaniesWorked+OverTime+PercentSalaryHike+PerformanceRating+RelationshipSatisfaction+StockOptionLevel+TotalWorkingYears+TrainingTimesLastYear+WorkLifeBalance+YearsAtCompany+YearsInCurrentRole+YearsSinceLastPromotion+YearsWithCurrManager,  
 data=trainnew,  
 hidden=c(8,5),  
 threshold=0.01,  
 err.fct="sse",  
 linear.output=FALSE,  
 stepmax=1500,rep=1,learningrate=0.5,  
 lifesign="full",  
 lifesign.step=100,  
 algorithm="backprop")

## hidden: 8, 5 thresh: 0.01 rep: 1/1 steps: 2 error: 159 time: 0.04 secs

summary(trainnew.nnbp)

## Length Class Mode   
## call 13 -none- call   
## response 2058 -none- numeric   
## covariate 61740 -none- numeric   
## model.list 2 -none- list   
## err.fct 1 -none- function  
## act.fct 1 -none- function  
## linear.output 1 -none- logical   
## data 31 data.frame list   
## net.result 1 -none- list   
## weights 1 -none- list   
## startweights 1 -none- list   
## generalized.weights 1 -none- list   
## result.matrix 302 -none- numeric

###### The training process needed 2 steps until all absolute partial derivatives of the error function were smaller than 0.01 (the default threshold).

###### Plotting the trained Neural Network

plot(trainnew.nnbp)

###### Confusion Matrix for Train

library(lattice)  
library(ggplot2)  
library(caret)  
library(e1071)  
  
trainnew$Prob = trainnew.nnbp$net.result[[1]]  
trainnew$Class = ifelse(trainnew$Prob>0.22,1,0)  
cm.emp = table(as.factor(trainnew$Class),as.factor(trainnew$Attrition))  
cm.emp

##   
## 1 2  
## 1 1740 318

###### Error Computation

sum((trainnew$Attrition - trainnew$Prob)^2) / 2

## [1] 159

###### The output of the computation matches to the error in neuralnet results

**Confusion Matrix for Test**

testnew\_prediction1=testnew[-c(31)]  
  
compute.output=neuralnet::compute(trainnew.nnbp,testnew\_prediction1)  
  
testnew$Predict.score = compute.output$net.result  
  
testnew$Class = ifelse(testnew$Predict.score>0.22,1,0)  
cm3=table(as.factor(testnew$Class),as.factor(testnew$Attrition))  
  
cm3

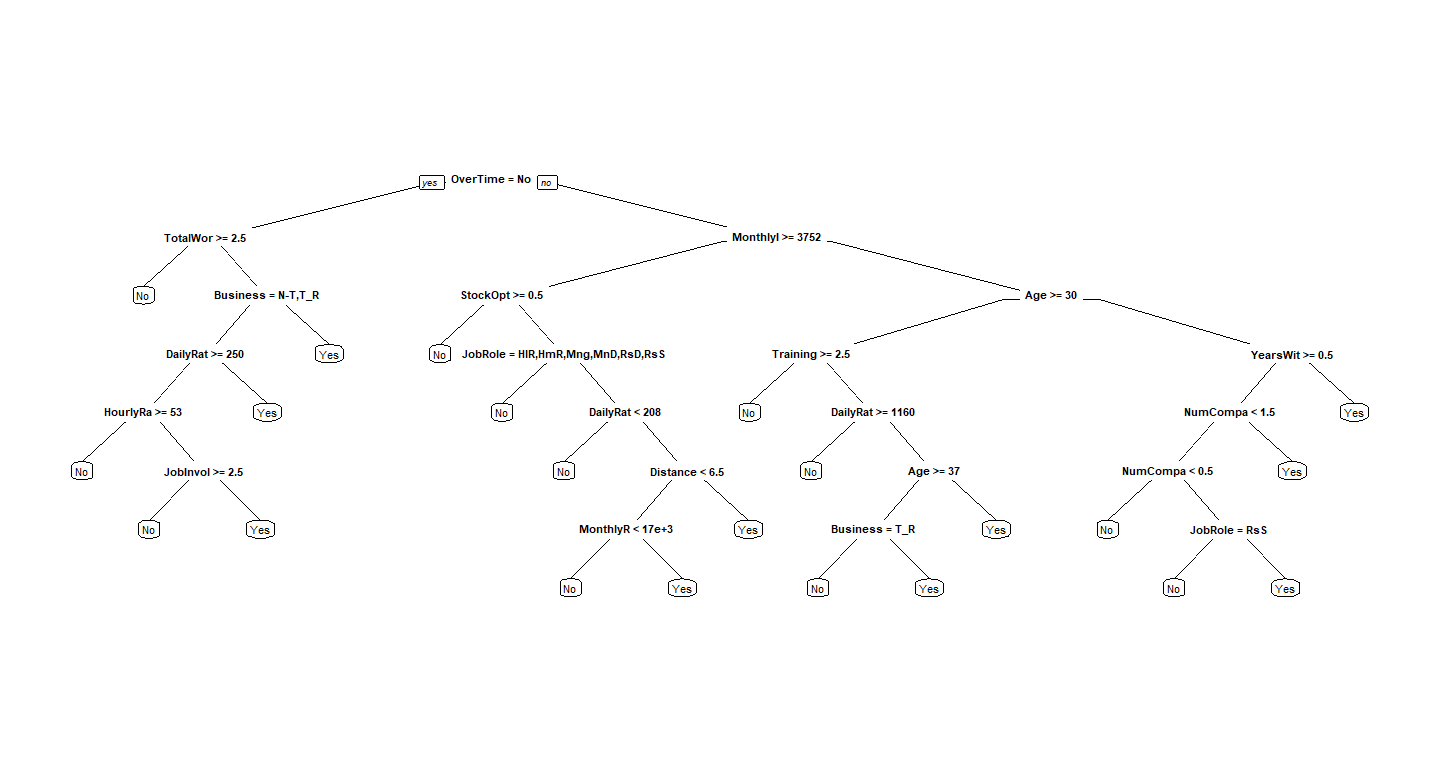
##   
## 1 2  
## 1 726 156

#(cm3[1]+cm3[4])/882 #confusion matrix for Test

##### 

## Build CART Model

library(rpart)  
library(rpart.plot)  
modelCart = rpart(Attrition ~ ., data=train, method="class")  
  
#Plotting the model  
prp(modelCart)



###### Predicting the model

predcart = predict(modelCart, newdata = test, type = "class")  
  
  
table(predcart)

## predcart  
## No Yes   
## 803 79

###### To get the accuracy of the CART model, we are doing confusion Matrix and accuracy test

cm1 = table(test$Attrition, predcart)  
cm1

## predcart  
## No Yes  
## No 709 17  
## Yes 94 62

#Accuracy of the model  
(cm1[1]+cm1[4])/(nrow(test))

## [1] 0.8741496599

###### The CART model has improved the accuracy but not by much. We will let do pruning as it did improve the accuracy in the last assignment.

## Validate CART Model

###### **Purning Code**

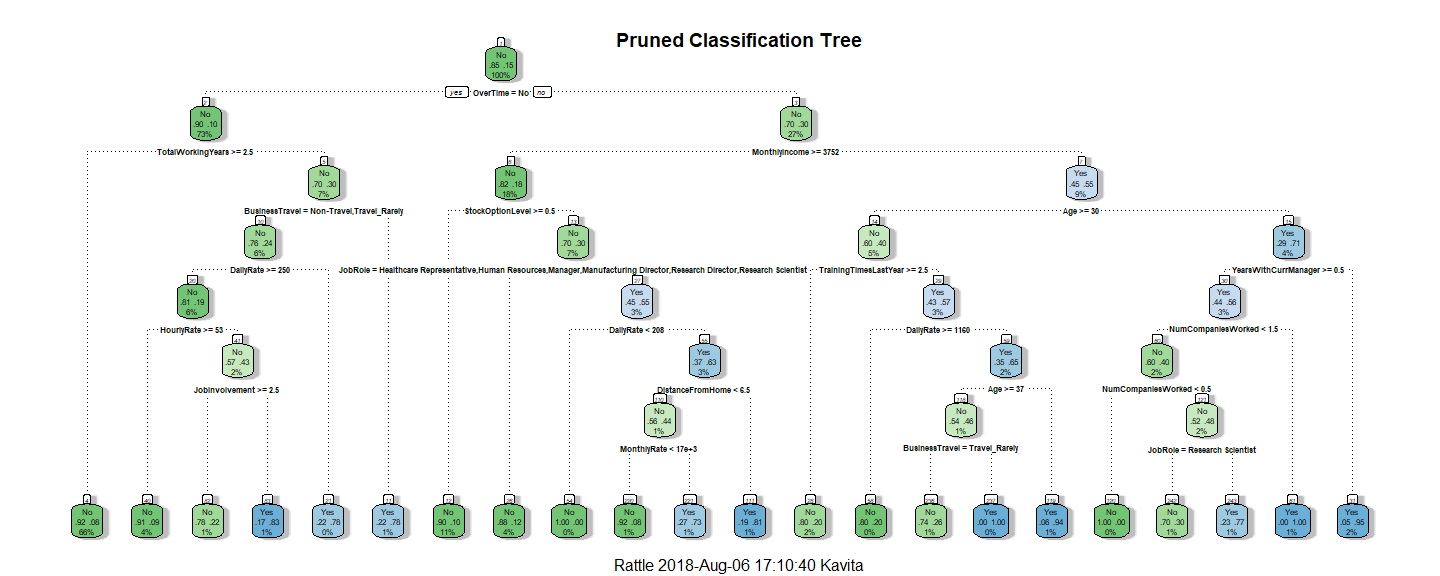
library(rattle)

## Rattle: A free graphical interface for data science with R.  
## Version 5.1.0 Copyright (c) 2006-2017 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

emp.ptree= prune(modelCart, cp= 0.001 ,"CP")  
printcp(emp.ptree)

##   
## Classification tree:  
## rpart(formula = Attrition ~ ., data = train, method = "class")  
##   
## Variables actually used in tree construction:  
## [1] Age BusinessTravel DailyRate   
## [4] DistanceFromHome HourlyRate JobInvolvement   
## [7] JobRole MonthlyIncome MonthlyRate   
## [10] NumCompaniesWorked OverTime StockOptionLevel   
## [13] TotalWorkingYears TrainingTimesLastYear YearsWithCurrManager   
##   
## Root node error: 318/2058 = 0.15451895  
##   
## n= 2058   
##   
## CP nsplit rel error xerror xstd  
## 1 0.031446541 0 1.00000000 1.00000000 0.051563024  
## 2 0.022012579 3 0.87735849 0.97798742 0.051094816  
## 3 0.018867925 4 0.85534591 0.95911950 0.050686356  
## 4 0.015723270 5 0.83647799 0.95283019 0.050548709  
## 5 0.014675052 8 0.78930818 0.94968553 0.050479602  
## 6 0.014150943 13 0.71383648 0.94025157 0.050271137  
## 7 0.012578616 15 0.68553459 0.92138365 0.049848990  
## 8 0.011006289 19 0.63522013 0.92767296 0.049990486  
## 9 0.010000000 21 0.61320755 0.92767296 0.049990486

fancyRpartPlot(emp.ptree, uniform=TRUE, main="Pruned Classification Tree")



## Ensembling Model

##### We will now develop the ensemble model for the two models - NN and CART to check if it improves the accuracy

predictions = data.frame(predictionCart= predcart,  
 predictionNN = predict.emp.nn)  
  
predictions$predictionEnsemble = as.factor(ifelse(predictions$predictionCart=='Yes'&predictions$predictionNN=='Yes','Yes','No'))  
  
#Confusion Matrix  
cm2 = table(test$Attrition, predictions$predictionEnsemble)  
cm2

##   
## No Yes  
## No 725 1  
## Yes 108 48

#Ensembeling Accuracy  
(cm2[1]+cm2[4])/(nrow(test))

## [1] 0.8764172336

###### The ensembling accuracy (88%) has been nearly close to the both NN (~90%) and CART (~88%) . Thus, ensembling may not be useful in all circumstances.

## Hypothesis Validation

###### 1. Overtime is the singlemost important factor contributing towards attrition.

###### Overtime remains the singlemost important factor contributing towards attrition in the CART model.

###### 2. Young people with lesser years at the company contribute to the attrition.

###### 3. Age and TotalWorkingYears are amongst the top 4th and 2nd factors influencing attrition in the random forest model.

## Ensembling models have improved accuracy

###### Ensembling models has not really helped improve the accuracy. For further more understanding, we should check the correlation between the predictions from the 2 models, usually uncorelated predictions improve accuracy. Stacking can be other option to be tried in case of larger dataset.