HR ANALYTICS: EMPLOYEE ATTRITION

BUAN 6356

Abstract

Human Resource Analytics using R

Vinay Kumar Singh, Megha Syam, Atul Kumar Verma and Zhengyu Wang

HR Analytics

Submitted by
Group 11

Under the guidance of
Sourav Chatterjee

The University of Texas at Dallas
Richardson, TX 75080
Dec 2018

ACKNOWLEDGEMENT

Our project on **HR Analytics** has been a great learning experience. We were exposed to a variance of subject matter, concerns and arguments that helped us collectively assemble and shape the project.

We acknowledge Sourav Chatterjee under whose guidance we were able to complete the project and effectively present its valuable benefits.

A greater share of inputs and knowledge from **each one of us** made this project report possible to its rightful accuracy.

To all our colleagues who have helped us either directly or indirectly, we are grateful for their valuable inputs.

TABLE OF CONTENTS

INDEX

LITERATURE

A substantial amount of money, time and training efforts are put together every year for recruiting the right people for a set of projects. A lot of time is invested on the employee before they start generating return for the organizations. With the advent of Data Analytics, HR want to explore more about their employee behavior to reduce the expenditure on recruitment and training. Analytics will also help to plan or allocate the task for the future work.

We will be using advanced Machine Learning techniques for predicted the customer who will are getting Attrition from the organization. The goal is to achieve maximum accuracy of predicting the behavior of the employee which in-directly reduces the expenditure for organization.

Retaining key employees is a major stake for any organization. But are there reliable ways to figure out if and why the best and most experienced employees are leaving prematurely? Most firms these days are already integrating the benefits of using analytics to introduce special efforts in regaining employees as well as hiring decisions. Lot of factors play key role in identifying significant predictors offering insights and meaning that can be interpreted using a statistical model language like R.

We have used the dataset provided by IBM related to HR Analytics in Kaggle.

HR Analytics

A very interesting branch of Analytics which is at an initial stage of using data to streamline process and application which are used in an organization. Human resources specialists are responsible for recruiting, screening, interviewing and placing workers. They may also handle employee relations, payroll, benefits, and training. Human resources managers plan, direct and coordinate the administrative functions of an organization (Wikipedia). Analytics and Data can help HR gather information about the employee sentiment with the company, can help transform the entire process of recruiting people in an organization along with many other examples.

Overview

Organization spend a lot of money, time and resources on hiring the right set of people fitting their work space. They also spend a lot money in training program for the employee so that they fit well with the organization and to increase the effectiveness of the employee. Hence it is very important for the Human resources to identify the people who will be leaving the company at the right point of time to identifying potential budget required for future process. It also helps reduce expenditure a firm is making on Human resource department.

Human Resource (HR) Analytics is an area in the field of analytics referring to the use of data and algorithm by the Human resource department to help improve employee performance and get better return on investment. It deals with the idea of generating valuable insight and decision to the Human resource department by working on employee and organization data for increasing efficiency & productivity for an organization.

Problem

Attrition in a company signifies reduction in staff and employee with the organization through various forms such as retirement, resignation, loss of client or any other. Our problem corresponds to the problem of identifying potential employee behavior of leaving/staying in an organization based on a 35-metrics gathered from Kaggle provided by IBM HR Analytics Employee & Attrition data-set.

Data Exploration

Columns: 35
Rows: 1470
Target Variable: Attrition
Missing Values: None

Description of Metadata:

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’

A glance at top 6 observations in the dataset:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount
1	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1
2	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1
3	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1
4	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1
5	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1
6	32	No	Travel_Frequently	1005	Research & Development	2	2	Life Sciences	1

EmployeeNumber	EnvironmentSatisfaction	Gender	HourlyRate	JobInvolvement	JobLevel	JobRole	JobSatisfaction	MaritalStatus
1	2	Female	94	3	2	Sales Executive	4	Single
2	3	Male	61	2	2	Research Scientist	2	Married
4	4	Male	92	2	1	Laboratory Technician	3	Single
5	4	Female	56	3	1	Research Scientist	3	Married
7	1	Male	40	3	1	Laboratory Technician	2	Married
8	4	Male	79	3	1	Laboratory Technician	4	Single

MonthlyIncome	MonthlyRate	NumCompaniesWorked	Over18	OverTime	PercentSalaryHike	PerformanceRating	RelationshipSatisfaction
5993	19479	8	Y	Yes	11	3	1
5130	24907	1	Y	No	23	4	4
2090	2396	6	Y	Yes	15	3	2
2909	23159	1	Y	Yes	11	3	3
3468	16632	9	Y	No	12	3	4
3068	11864	0	Y	No	13	3	3

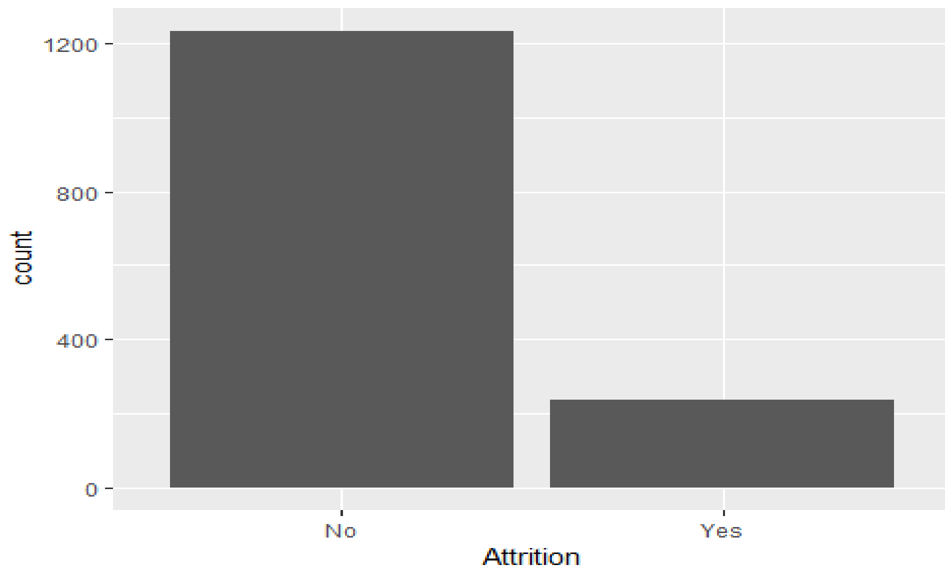
StandardHours	StockOptionLevel	TotalWorkingYears	TrainingTimesLastYear	WorkLifeBalance	YearsAtCompany	YearsInCurrentRole
80	0	8	0	1	6	4
80	1	10	3	3	10	7
80	0	7	3	3	0	0
80	0	8	3	3	8	7
80	1	6	3	3	2	2
80	0	8	2	2	7	7

YearsSinceLastPromotion	YearsWithCurrManager
0	5
1	7
0	0
3	0
2	2
3	6

EXPLORATORY DATA ANALYIS

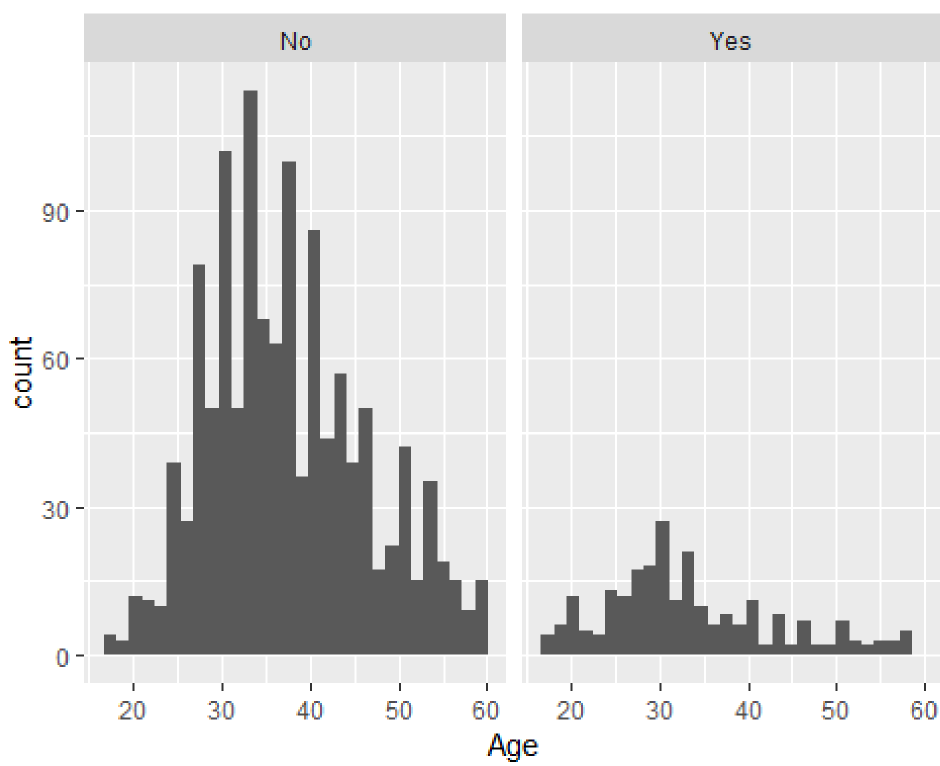
Attrition distribution

```
a <-ggplot(hr, aes(x= Attrition)) +
  geom_bar()
a+ labs(x="Attrition")
```



Plot of Age faceted by Attrition

```
ggplot(hr, aes(x= Age)) +
  geom_histogram() + facet_wrap(~ Attrition)
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```r
options(scipen=999)
E <- hr %>%
  group_by(Attrition) %>%
  summarise(median(Age))
E
```

```
## # A tibble: 2 x 2
##   Attrition `median(Age)`
##   <fct>             <int>
## 1 No                   36
## 2 Yes                  32
```

Inference : From both the plot and table it can be inferred that employees aged between 30- 35 tend to leave the company more.

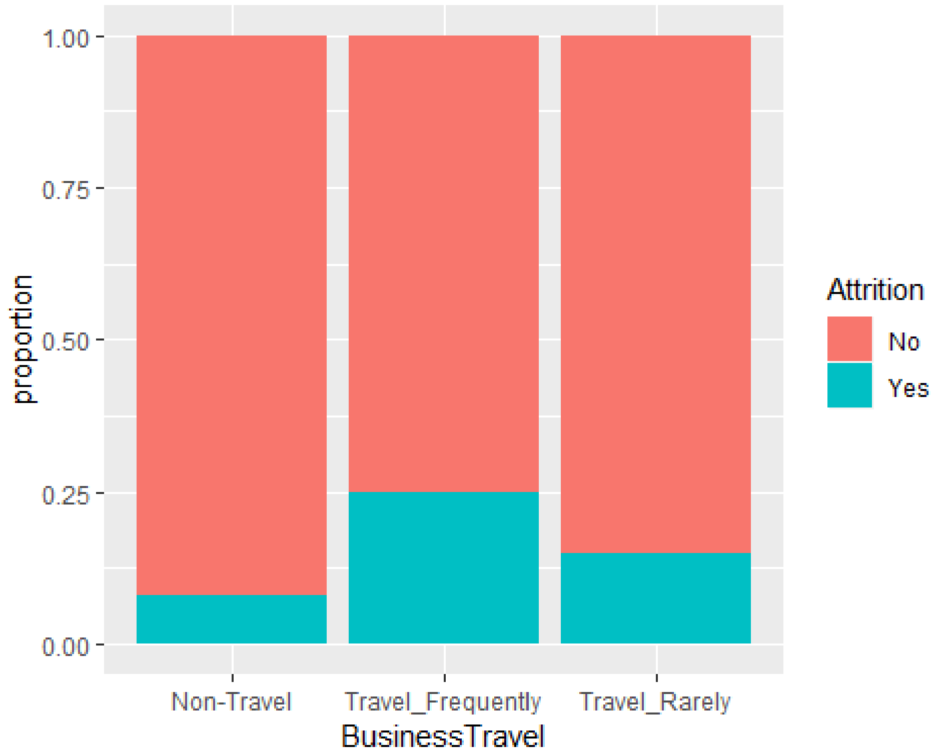
Proportion of Attrition by department

```r
ggplot(hr, aes(x=Department, fill = Attrition)) + geom_bar(position = "fill") +
  ylab("proportion")
```

Inference: Employees from sales department tend to leave the company more.

Proportion of Attrition by Business Travel

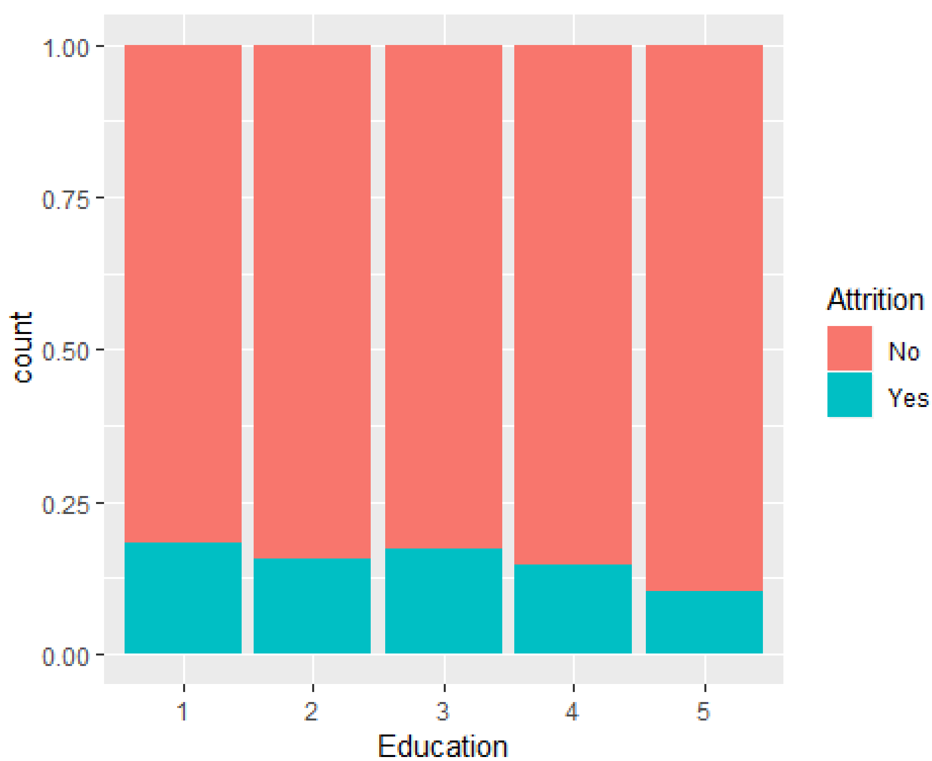
```r
ggplot(hr, aes(x=BusinessTravel, fill = Attrition)) + geom_bar(position = "fill") +
  ylab("proportion")
```



Inference: Employees who travel more frequently tend to leave the company more.

Proportion of Attrition by Education Level

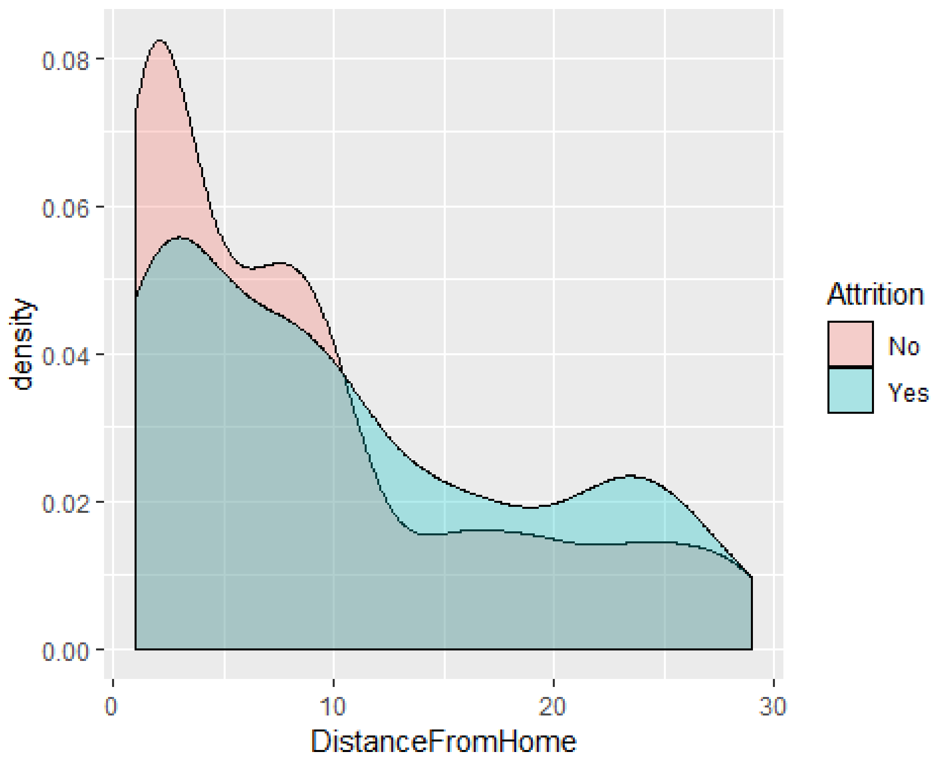
```
ggplot(hr, aes(x=Education, fill = Attrition)) + geom_bar(position = "fill")
```



Inference: Employees who are below college level tend to leave the company more.

Attrition vs Distance from home

```r
ggplot(hr, aes(x=DistanceFromHome, fill = Attrition)) + geom_density(alpha=0.3)
```

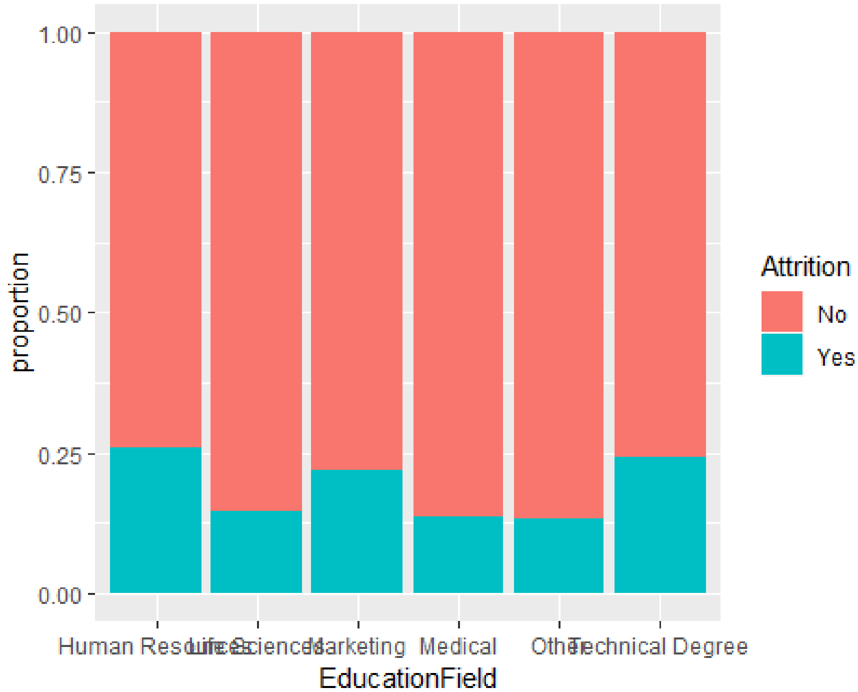



```r
options(scipen=999)
D <- hr %>%
  group_by(Attrition) %>%
  summarise(mean(DistanceFromHome))
D

## # A tibble: 2 x 2
##   Attrition `mean(DistanceFromHome)`
##   <fct>                        <dbl>
## 1 No                            8.92
## 2 Yes                          10.6
```

Inference: More employees stay near their office. As shown in the graph, as the distance increase attrition increase. Also, as shown in the table, the average distance from home is higher (10.6 miles) for employees who leave the office.

Proportion of Attrition by Education

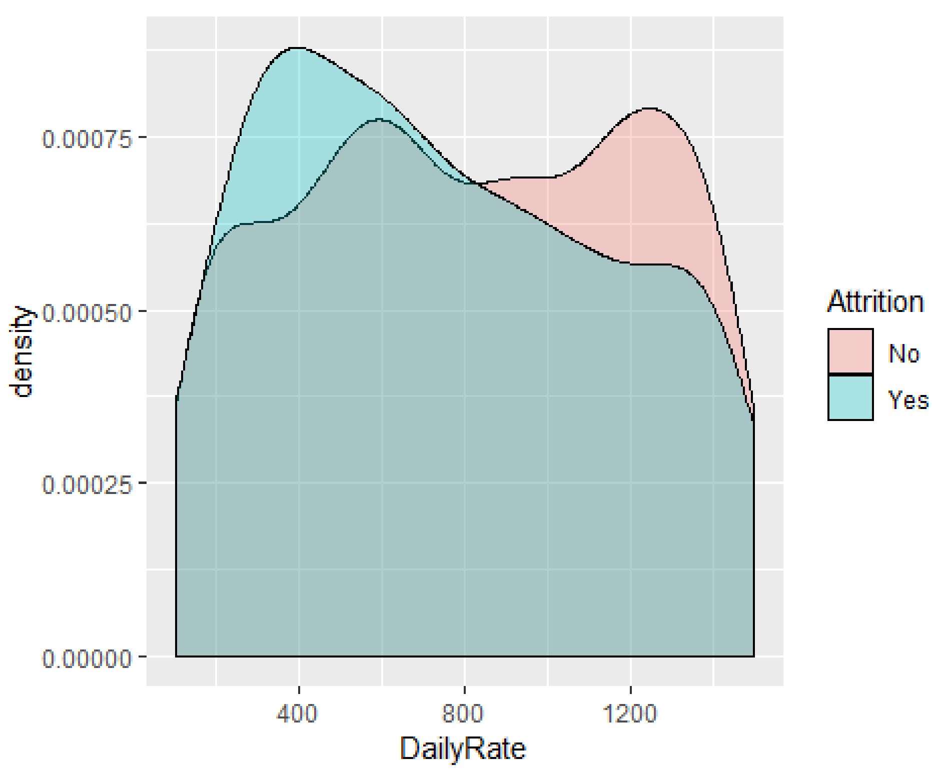
```r
ggplot(hr, aes(x=EducationField, fill = Attrition)) + geom_bar(position = "fill") +
ylab("proportion")
```



Inference: Employees with hr as their education tend to leave the company more.

Daily rate vs Attrition

```
ggplot(hr, aes(x= DailyRate, fill= Attrition)) + geom_density(alpha = 0.3)
```




```
options(scipen=999)
B <- hr %>%
  group_by(Attrition) %>%
```

```
  summarise(mean(DailyRate))
B
```

```
## # A tibble: 2 x 2
##   Attrition `mean(DailyRate)`
##   <fct>                 <dbl>
## 1 No                     813.
## 2 Yes                    750.
```

Inference: As shown in the graph, employees with less daily rate leave the company more and employees with high daily rate tend to leave less. Also, the table shows that the employees who leave the company have less average daily rate.

Breakdown of dailyrate by department

```
ggplot(hr, aes(x= DailyRate, fill= Department)) + geom_density(alpha = 0.3)
```


```
options(scipen=999)
A <- hr %>%
  group_by(Department) %>%
  summarise(mean(DailyRate), min(DailyRate), max(DailyRate))
A
```

```
## # A tibble: 3 x 4
##   Department             `mean(DailyRate~ `min(DailyRate)` `max(DailyRate)`
##   <fct>                             <dbl>            <dbl>            <dbl>
## 1 Human Resources                    752.              106             1444
## 2 Research & Developme~              807.              102             1496
## 3 Sales                              800.              107             1499
```

Inference: Human Resources department's daily rate is less compared to other departments. Research and Development department has highest average daily rate.

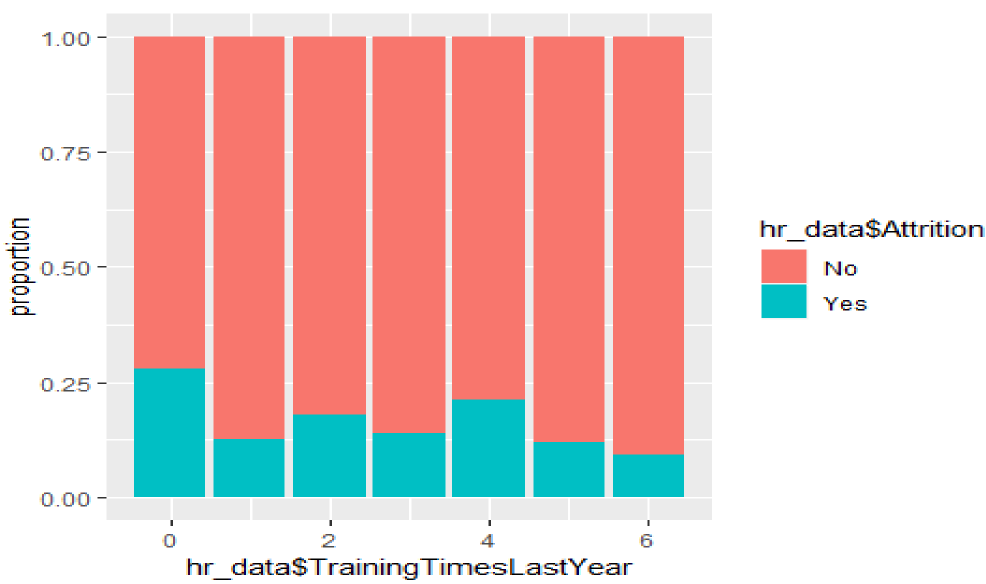
Breakdown of dailyrate by Education field

```
options(scipen=999)
C <- hr %>%
  group_by(EducationField) %>%
  summarise(mean(DailyRate), min(DailyRate), max(DailyRate))
C

## # A tibble: 6 x 4
##   EducationField   `mean(DailyRate)` `min(DailyRate)` `max(DailyRate)`
##   <fct>                        <dbl>            <dbl>            <dbl>
## 1 Human Resources               675.              106             1420
## 2 Life Sciences                 804.              102             1498
## 3 Marketing                     728.              118             1499
## 4 Medical                       823.              109             1495
## 5 Other                         796.              116             1474
## 6 Technical Degree              842.              107             1496
```

Inference: Employees with a technical degree have highest average daily rate and the employee from Marketing has highest daily rate.

Proportion of Attrition by TrainingTimesLastYear

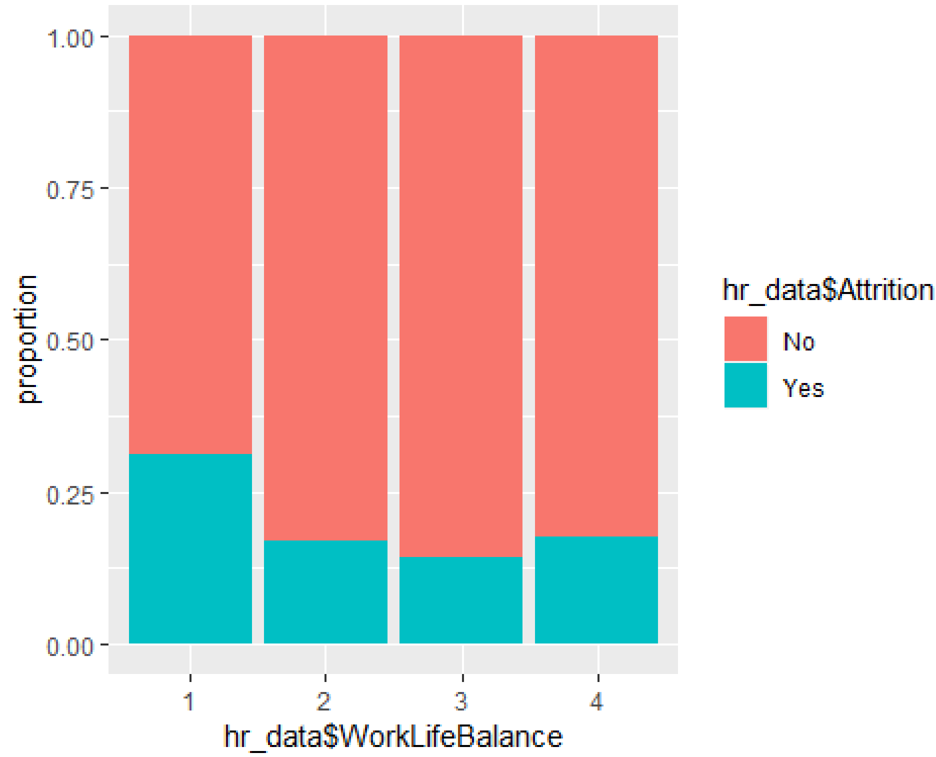
```
ggplot(hr_data, aes(x=hr_data$TrainingTimesLastYear, fill = hr_data$Attrition)) + geom_ba
r(position = "fill") +
  ylab("proportion")
```



Inference: Employees who weren't trained last year(2016) left the company more.

Proportion of Attrition by WorkLifeBalance

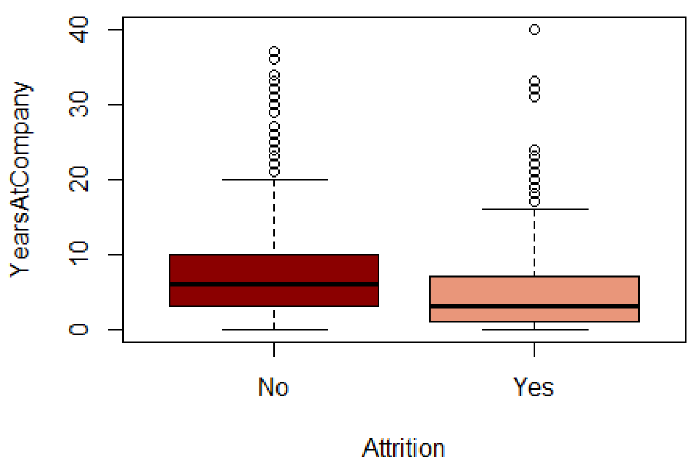
```
ggplot(hr_data, aes(x=hr_data$WorkLifeBalance, fill = hr_data$Attrition)) + geom_bar(posi
tion = "fill") +
  ylab("proportion")
```



Inference: Employees who had bad work life balance left the company more.

Box Plot of Attriton by YearsAtCompany

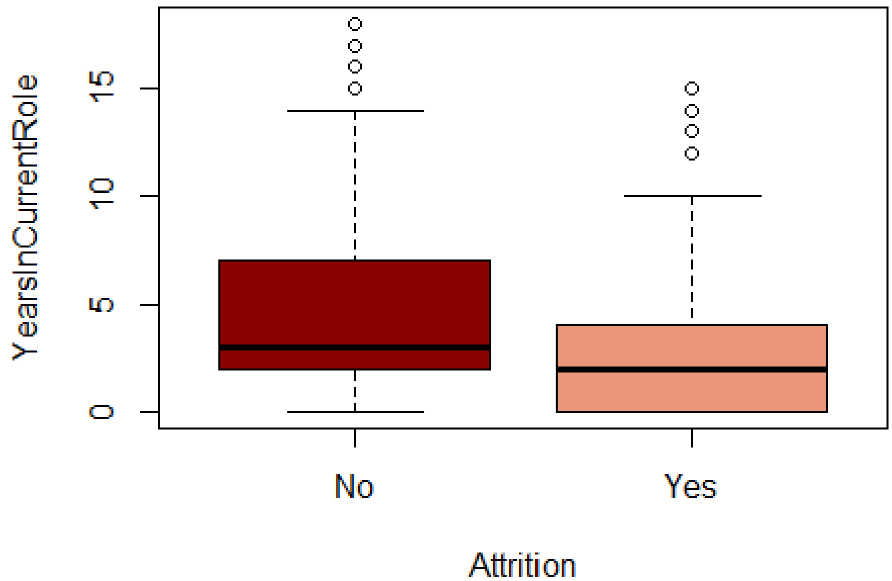
```
plot(YearsAtCompany~Attrition,data = hr_data,col=colors()[100:102])
```



Inference: Employees who left the company had fewer average years of experience at that company than people who didn't leave.

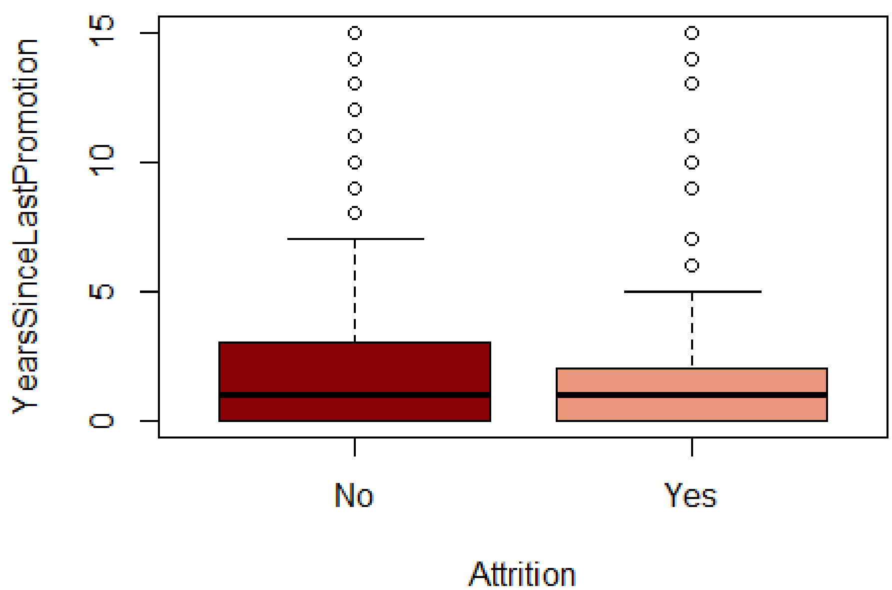
Box Plot of Attriton by CurrentRole

```
plot(YearsInCurrentRole~Attrition,data = hr_data,col=colors()[100:102])
```



Box Plot of Attriton by YearsSinceLastPromotion

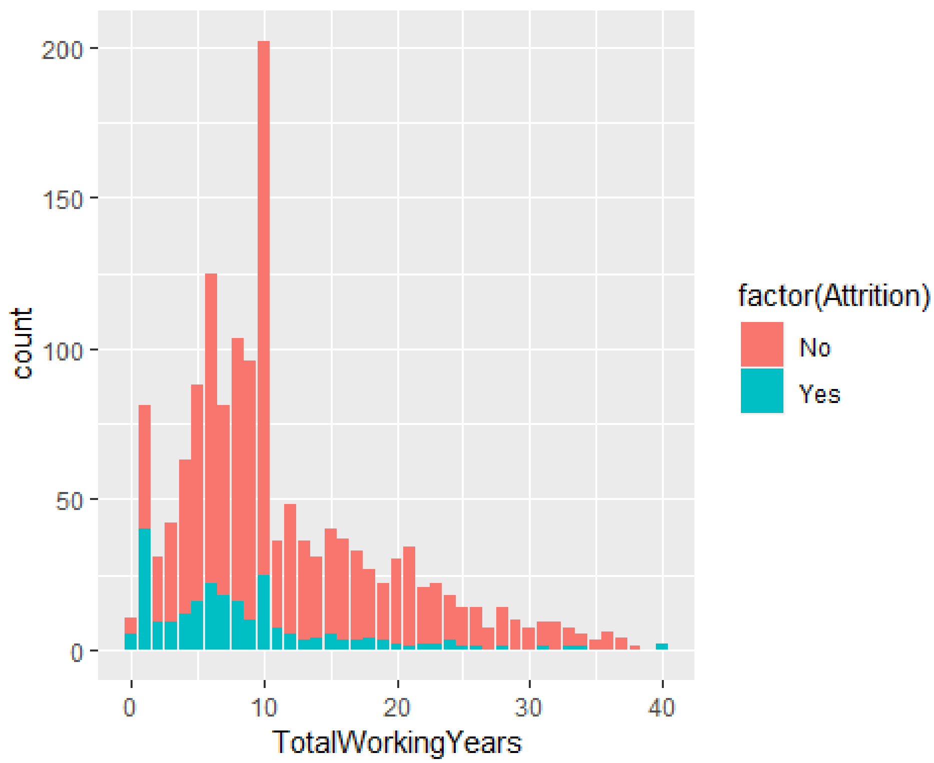
```
plot(YearsSinceLastPromotion~Attrition,data = hr_data,col=colors()[100:102])
```



Count of Attrition by TotalWorkingYears

Majority of Employees have 0-10 years as Total working years. As number of years increases, attrition of No increases.

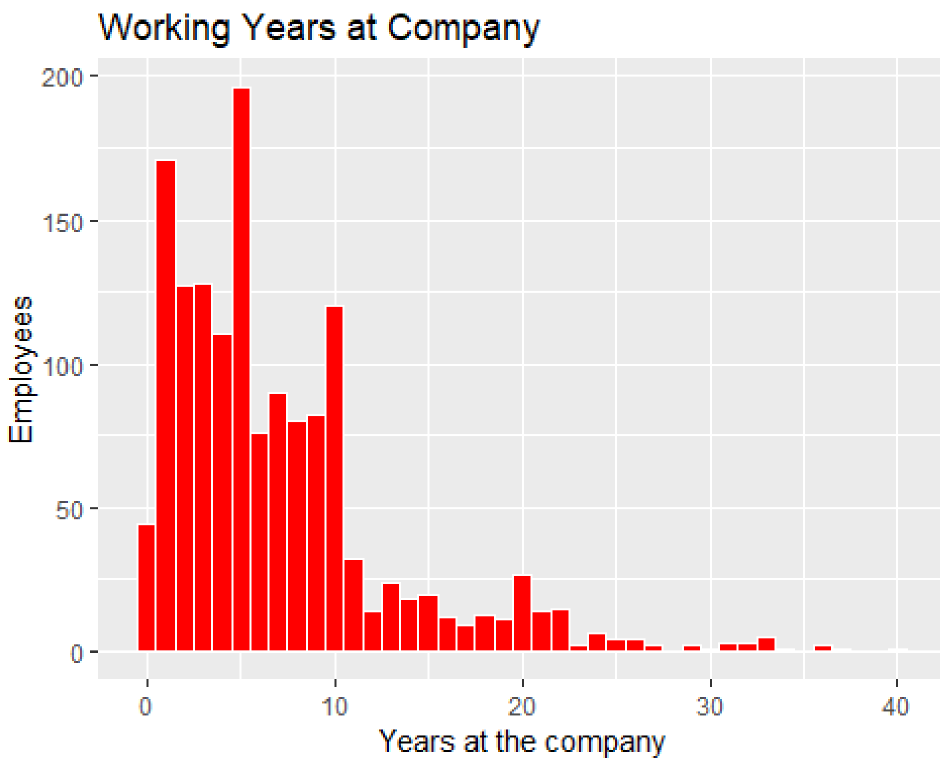
```
plot_1 = ggplot(hr_data, aes(TotalWorkingYears,fill = factor(Attrition)))
plot_2 = plot_1 + geom_histogram(stat="count")

## Warning: Ignoring unknown parameters: binwidth, bins, pad

print(plot_2)
```



Plot of Employees by YearsAtCompany

Most of the employees are new and have served the company for less than 10 years.

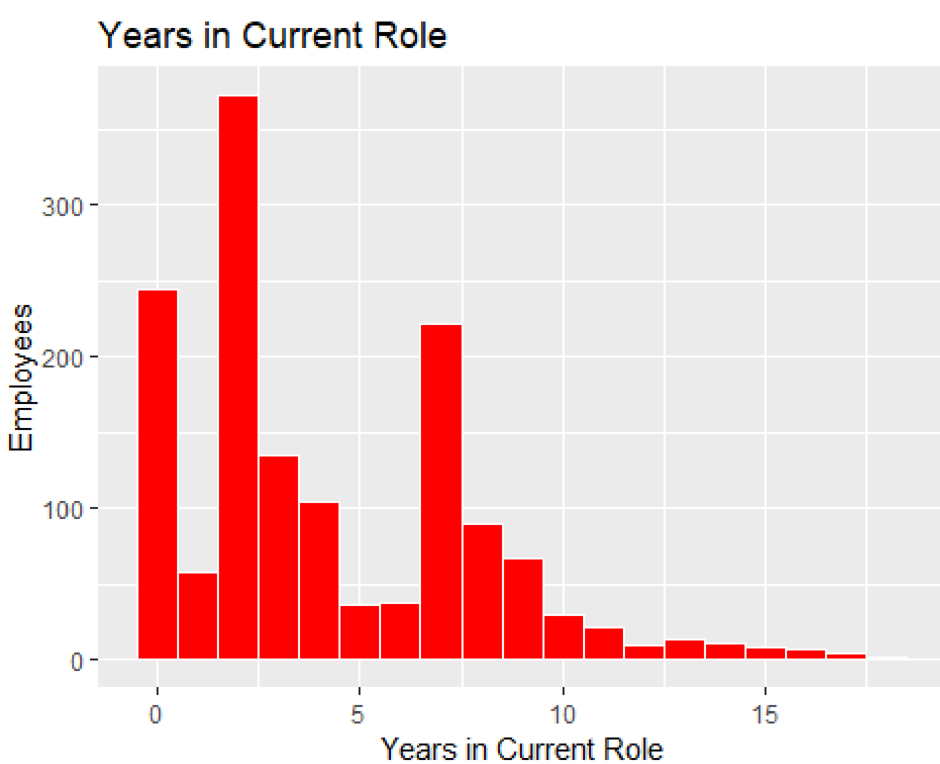
```
ggplot(hr_data) +
  geom_histogram(mapping=(aes(YearsAtCompany)),fill="red",col="white",binwidth = 1) +
  labs(x="Years at the company", y="Employees", title="Working Years at Company") + theme
(legend.position="none")
```



Plot of Employees Count by CurrentRole

Years that Majority of employees remain in the current role are between 0-7 years. Most of the employees have been in the same role for long period.

```
ggplot(hr_data) +
  geom_histogram(mapping=(aes(YearsInCurrentRole)),fill="red",col="white",binwidth = 1) +
  labs(x="Years in Current Role", y="Employees", title="Years in Current Role") + theme(l
egend.position="none")
```



Algorithms for Attrition Prediction:

Logistic Regression

How & why to choose it:

Logistic regression is highly popular and powerful in terms of classification. Similar with linear regression, it relies on a specific model relating the predictors with the outcome. Since we must specify the predictors and include their form in this algorithm, even small datasets can be used for building logistic regression classifiers, which is the case here.

Brief description of the algorithm:

The idea behind logistic regression is straightforward: instead of using Y directly as the outcome variable, we use a function of it as the logit (the log of odds), which can be modeled as a linear function of predictors. Once the logit has been predicted, it can be mapped back to a probability.

Data Preprocessing:

1.We drop the obvious needless columns here: Employee Count, Over 18, Employee Number (all the same), StandardHours (all the same)

```
hr.df <- hr.df[, -c(9,10,22,27)]
```

2. treat the below variables as categorical

```
hr.df$Education <- factor(hr.df$Education,
                          levels = c(1,2,3,4,5),
          labels = c('Below College','College','Bachelor','Master','Doctor'))

hr.df$EnvironmentSatisfaction <- factor(hr.df$EnvironmentSatisfaction,
                                        levels = c(1,2,3,4),
          labels = c('Low','Medium','High','Very High'))

hr.df$JobInvolvement <- factor(hr.df$JobInvolvement,
                                        levels = c(1,2,3,4),
                                      labels = c('Low','Medium','High','Very High'))

hr.df$JobLevel <- factor(hr.df$JobLevel,
                               levels = c(1,2,3,4,5),
                      labels = c('Very Low','Low','Medium','High','Very High'))

hr.df$JobSatisfaction <- factor(hr.df$JobSatisfaction,
                               levels = c(1,2,3,4),
                               labels = c('Low','Medium','High','Very High'))

hr.df$PerformanceRating <- factor(hr.df$PerformanceRating,
                                  levels = c(1,2,3,4),
      labels = c('Low','Good','Excellent','Outstanding'))

hr.df$RelationshipSatisfaction <- factor(hr.df$RelationshipSatisfaction,
                                  levels = c(1,2,3,4),
                           labels = c('Low','Medium','High','Very High'))

hr.df$WorkLifeBalance <- factor(hr.df$WorkLifeBalance,
                                         levels = c(1,2,3,4),
```

```
                                labels = c('Bad','Good','Better','Best'))

hr.df$StockOptionLevel <- factor(hr.df$StockOptionLevel,
                                 levels = c(0,1,2,3),
                                 labels = c('Low','Medium','High','Very High'))
```

3. partition data into training and validation dataset: 60% of training, 40% of validation

```
library(caret)
training.index <- createDataPartition(hr.df$Attrition, p = 0.60, list = FALSE)

hr.train.df <- hr.df[training.index, ]
hr.valid.df <- hr.df[-training.index, ]
```

4. normalize the data for numeric variables, since they are not using the same metrics (e.g. years vs dollars)

```
hr.norm <- preProcess(hr.train.df, method = c("center", "scale"))

hr.train.norm <- predict(hr.norm, hr.train.df)
hr.valid.norm <- predict(hr.norm, hr.valid.df)
```

Now we use the normalized data to run logistic regression

```
lm.fit <- glm(Attrition~., data = hr.train.norm, family = "binomial")
```

show coefficients and odds:

```
lm.summary <- data.frame(summary(lm.fit)$coefficients, odds = exp(coef(lm.fit)))
options(scipen = 999)
round(lm.summary, 5)

##                                      Estimate Std..Error  z.value Pr...z..
## (Intercept)                         -12.23430  682.56490 -0.01792  0.98570
## ï..Age                               -0.20649    0.18092 -1.14134  0.25373
## BusinessTravelTravel_Frequently       1.39035    0.58388  2.38123  0.01725
## BusinessTravelTravel_Rarely           0.69725    0.52543  1.32702  0.18450
## DailyRate                             0.01671    0.12840  0.13013  0.89646
## DepartmentResearch & Development     15.57862  682.56390  0.02282  0.98179
## DepartmentSales                      13.97253  682.56463  0.02047  0.98367
## DistanceFromHome                      0.61637    0.12986  4.74626  0.00000
## EducationCollege                      0.63280    0.50784  1.24606  0.21274
## EducationBachelor                     0.54407    0.45995  1.18288  0.23686
## EducationMaster                       0.80127    0.49071  1.63290  0.10249
## EducationDoctor                       0.84912    1.02464  0.82869  0.40728
## EducationFieldLife Sciences          -2.73969    1.23197 -2.22382  0.02616
## EducationFieldMarketing              -2.52796    1.29110 -1.95800  0.05023
## EducationFieldMedical                -2.74834    1.22632 -2.24113  0.02502
## EducationFieldOther                  -2.75929    1.32921 -2.07589  0.03790
## EducationFieldTechnical Degree       -1.12863    1.24834 -0.90411  0.36594
## EnvironmentSatisfactionMedium        -1.20783    0.41225 -2.92983  0.00339
## EnvironmentSatisfactionHigh          -1.70237    0.39495 -4.31036  0.00002
## EnvironmentSatisfactionVery High     -1.79475    0.38617 -4.64759  0.00000
## GenderMale                            0.44292    0.27971  1.58350  0.11331
## HourlyRate                            0.01372    0.13395  0.10243  0.91841
## JobInvolvementMedium                 -1.42499    0.52856 -2.69598  0.00702
## JobInvolvementHigh                   -1.39952    0.48350 -2.89456  0.00380
## JobInvolvementVery High              -2.41489    0.67939 -3.55448  0.00038
## JobLevelLow                          -2.05453    0.70200 -2.92669  0.00343
## JobLevelMedium                       -0.13703    1.05718 -0.12962  0.89687
```

```
## JobLevelHigh                        -1.31080    1.68926 -0.77596  0.43777
## JobLevelVery High                    2.39125    2.24811  1.06367  0.28748
## JobRoleHuman Resources              15.88309  682.56427  0.02327  0.98144
## JobRoleLaboratory Technician         0.59511    0.91329  0.65161  0.51465
## JobRoleManager                      -0.03609    1.35513 -0.02663  0.97876
## JobRoleManufacturing Director        0.72498    0.79195  0.91544  0.35996
## JobRoleResearch Director            -3.23853    1.70552 -1.89885  0.05758
## JobRoleResearch Scientist           -0.69021    0.94399 -0.73116  0.46468
## JobRoleSales Executive               3.52778    1.75317  2.01223  0.04420
## JobRoleSales Representative          3.40382    1.86366  1.82642  0.06779
## JobSatisfactionMedium               -0.76344    0.39895 -1.91363  0.05567
## JobSatisfactionHigh                 -1.10214    0.37036 -2.97583  0.00292
## JobSatisfactionVery High            -1.47204    0.37674 -3.90733  0.00009
## MaritalStatusMarried                 0.56375    0.40535  1.39079  0.16429
## MaritalStatusSingle                  0.79189    0.56980  1.38977  0.16460
## MonthlyIncome                       -0.46236    0.60624 -0.76267  0.44566
## MonthlyRate                         -0.00410    0.12954 -0.03164  0.97476
## NumCompaniesWorked                   0.41876    0.14121  2.96550  0.00302
## OverTimeYes                          2.68211    0.31297  8.56997  0.00000
## PercentSalaryHike                   -0.12532    0.19737 -0.63496  0.52545
## PerformanceRatingOutstanding         0.26390    0.58663  0.44986  0.65281
## RelationshipSatisfactionMedium      -1.55728    0.45655 -3.41100  0.00065
## RelationshipSatisfactionHigh        -0.90175    0.36680 -2.45843  0.01395
## RelationshipSatisfactionVery High   -0.88901    0.35844 -2.48021  0.01313
## StockOptionLevelMedium              -1.41308    0.43783 -3.22747  0.00125
## StockOptionLevelHigh                -1.86264    0.69165 -2.69304  0.00708
## StockOptionLevelVery High           -0.79418    0.70317 -1.12943  0.25872
## TotalWorkingYears                   -0.38057    0.32921 -1.15604  0.24766
## TrainingTimesLastYear               -0.29043    0.14050 -2.06715  0.03872
## WorkLifeBalanceGood                 -0.73283    0.53047 -1.38148  0.16713
## WorkLifeBalanceBetter               -1.70897    0.51107 -3.34391  0.00083
## WorkLifeBalanceBest                 -0.68994    0.60451 -1.14132  0.25374
## YearsAtCompany                       0.41483    0.32741  1.26703  0.20515
## YearsInCurrentRole                  -0.64256    0.26742 -2.40286  0.01627
## YearsSinceLastPromotion              0.69437    0.21008  3.30521  0.00095
## YearsWithCurrManager                -0.47495    0.25373 -1.87184  0.06123
##                                             odds
## (Intercept)                              0.00000
## ï..Age                                   0.81343
## BusinessTravelTravel_Frequently          4.01624
## BusinessTravelTravel_Rarely              2.00823
## DailyRate                                1.01685
## DepartmentResearch & Development   5830566.79461
## DepartmentSales                    1170014.88448
## DistanceFromHome                         1.85219
## EducationCollege                         1.88287
## EducationBachelor                        1.72300
## EducationMaster                          2.22838
## EducationDoctor                          2.33758
## EducationFieldLife Sciences              0.06459
## EducationFieldMarketing                  0.07982
## EducationFieldMedical                    0.06403
## EducationFieldOther                      0.06334
## EducationFieldTechnical Degree           0.32347
## EnvironmentSatisfactionMedium            0.29884
```

```
## EnvironmentSatisfactionHigh                 0.18225
## EnvironmentSatisfactionVery High            0.16617
## GenderMale                                  1.55725
## HourlyRate                                  1.01382
## JobInvolvementMedium                        0.24051
## JobInvolvementHigh                          0.24672
## JobInvolvementVery High                     0.08938
## JobLevelLow                                 0.12815
## JobLevelMedium                              0.87194
## JobLevelHigh                                0.26960
## JobLevelVery High                          10.92717
## JobRoleHuman Resources                7905631.55555
## JobRoleLaboratory Technician                1.81324
## JobRoleManager                              0.96456
## JobRoleManufacturing Director               2.06469
## JobRoleResearch Director                    0.03922
## JobRoleResearch Scientist                   0.50147
## JobRoleSales Executive                     34.04825
## JobRoleSales Representative                30.07874
## JobSatisfactionMedium                       0.46606
## JobSatisfactionHigh                         0.33216
## JobSatisfactionVery High                    0.22946
## MaritalStatusMarried                        1.75726
## MaritalStatusSingle                         2.20757
## MonthlyIncome                               0.62979
## MonthlyRate                                 0.99591
## NumCompaniesWorked                          1.52008
## OverTimeYes                                14.61589
## PercentSalaryHike                           0.88221
## PerformanceRatingOutstanding                1.30200
## RelationshipSatisfactionMedium              0.21071
## RelationshipSatisfactionHigh                0.40586
## RelationshipSatisfactionVery High           0.41106
## StockOptionLevelMedium                      0.24339
## StockOptionLevelHigh                        0.15526
## StockOptionLevelVery High                   0.45195
## TotalWorkingYears                           0.68347
## TrainingTimesLastYear                       0.74794
## WorkLifeBalanceGood                         0.48055
## WorkLifeBalanceBetter                       0.18105
## WorkLifeBalanceBest                         0.50161
## YearsAtCompany                              1.51412
## YearsInCurrentRole                          0.52594
## YearsSinceLastPromotion                     2.00245
## YearsWithCurrManager                        0.62192
```

Interpret the results:

For illustration, the odds has been present using the EXP() function here. (the odds = e^coefficient) For continuous variables, the odds is the multiplicative factor by which the odds (of belonging to class 1) increase when the value of predictor is increased by 1 unit, holding all other predictors constant. For dummy variable predictors, the odds means the chance on outcome with predictor of being 1 vs.being zero.

As we can see, OvertimeYes, JobRoleSalesExecutive, JobRoleSalesRep, JoblevelVeryHigh and BusinessTravelTravel_Frequently has the largest odds here positively (the 3 variables with coefficient of 14 are not discussed here due to high p value), while other predictors have a small to moderate impact on attrition, either positively or negatively.

Evaluate the results on validation dataset using confusion matrix:

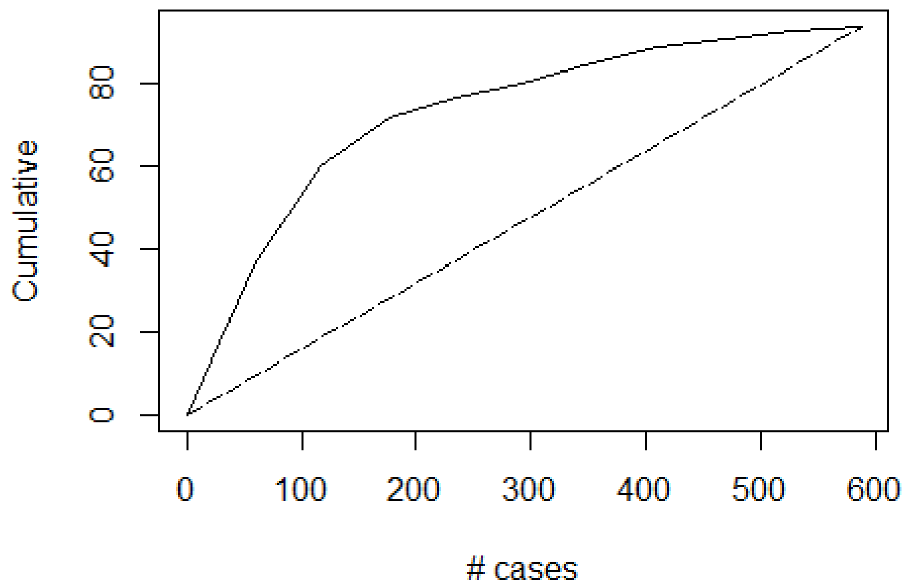
```
pred <- predict(lm.fit, hr.valid.norm, type = 'response')
confusionMatrix(as.factor(ifelse(pred > 0.5, "Yes", "No")),
                as.factor(hr.valid.norm$Attrition))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  461  49
##        Yes  32  45
##
##                Accuracy : 0.862
##                  95% CI : (0.8314, 0.8889)
##     No Information Rate : 0.8399
##     P-Value [Acc > NIR] : 0.07773
##
##                   Kappa : 0.4465
##  Mcnemar's Test P-Value : 0.07544
##
##             Sensitivity : 0.9351
##             Specificity : 0.4787
##          Pos Pred Value : 0.9039
##          Neg Pred Value : 0.5844
##              Prevalence : 0.8399
##          Detection Rate : 0.7853
##    Detection Prevalence : 0.8688
##       Balanced Accuracy : 0.7069
##
##        'Positive' Class : No
##
```

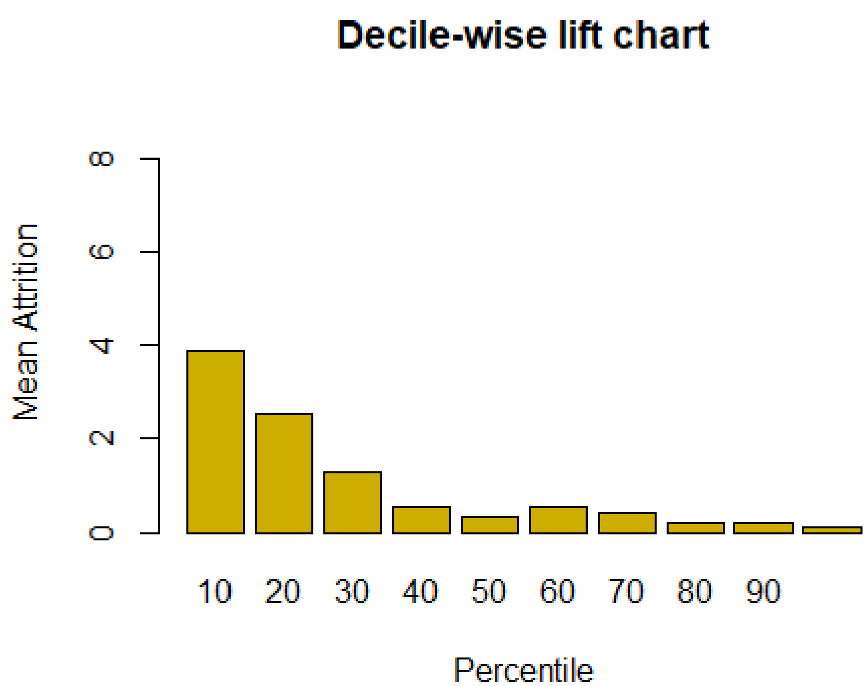
As we can see, the overall accuracy is 86.2% with a Specificity of 47.87%.

Plotting lift/decile chart:

```
library(gains)
hr.valid.norm$isAttrition <- 1 * (hr.valid.norm$Attrition == "Yes")
gain <- gains(hr.valid.norm$isAttrition, pred)

### Plot Lift Chart
plot(c(0,gain$cume.pct.of.total*sum(hr.valid.norm$isAttrition))~c(0,gain$cume.obs),
     xlab = "# cases", ylab = "Cumulative", main = "", type = "l")
lines(c(0,sum(hr.valid.norm$isAttrition))~c(0, dim(hr.valid.norm)[1]), lty = 5)
```




```
### Plot decile-wise chart
heights <- gain$mean.resp/mean(hr.valid.norm$isAttrition)
midpoints <- barplot(heights, names.arg = gain$depth,  ylim = c(0,9), col = "gold3",
                     xlab = "Percentile", ylab = "Mean Attrition",
                     main = "Decile-wise lift chart")
```



CART Model

Features:

- Data Driven method that can be used for both classification and prediction
- Creates splits on predictors using logical rules

Advantages:

- Very easy to interpret
- Creates interesting analysis on the predictors, which is very useful for decision making.
- It can reduce the dimension by Pruning (Cutting tree back).

Weakness:

- Requires large dataset to build a good classifier
- Splits are done on one predictor at a time rather than on combinations of predictors

The columns below are not giving any information, so these columns can be removed from the dataset.

- **EmployeeCount:** Values same for all the observations
- **EmployeeNumber**: Serial number for the observations
- **Over18:** Values same for all the observations.
- **StandardHours:** Values same for all the observations

```
#Removing columns that doesn't give useful information
hr <- hr[-c(9,10,22,27)]
```

Converting Yes and No in Attrition Column to 1 and 0 inorder to use gain function.

```
hr$Attrition <- as.numeric(hr$Attrition) -1
```

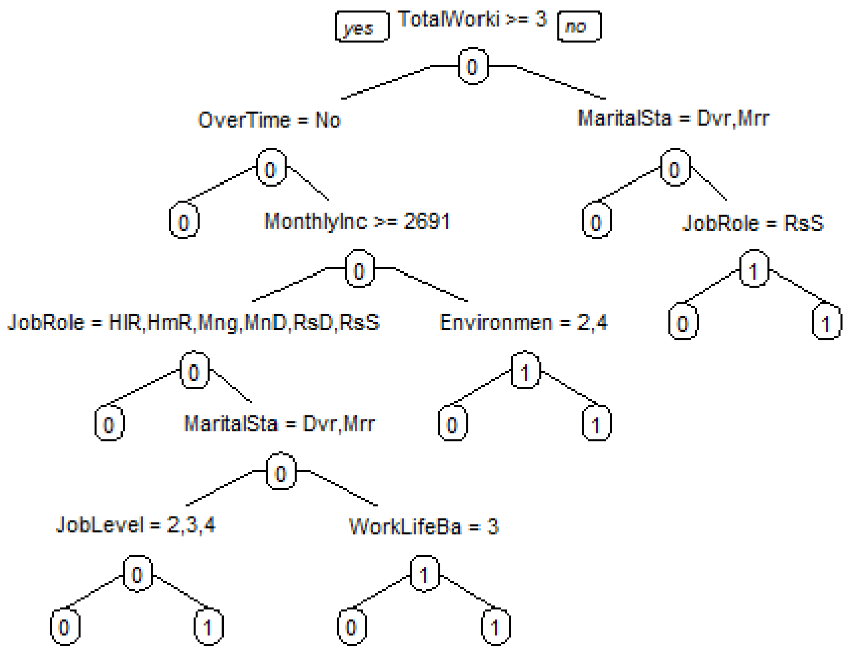
Splitting data into 60:40 Training and Validation Datasets:

```
#Data partition into training and validation datasets.

set.seed(111)
training.index <- createDataPartition(hr$Attrition, p = 0.6, list= FALSE)
hr.train <- hr[training.index, ]
hr.valid <- hr[-training.index, ]
```

Generating Classification Tree:

Running the model with rpart() function (Recursive Partitioning and Regression Trees).

```
### Generate classification tree
hrtree <- rpart( Attrition~ ., data = hr.train, method = "class")
prp(hrtree,type=1, split.font = 1, varlen = -10, cex= 0.7)
```



Count of leaves in fully grown Tree:

```
hrfulltree <- rpart(Attrition ~ ., data = hr.train,
                    method = "class", cp = 0, minsplit = 1)

length(hrfulltree$frame$var[hrfulltree$frame$var == "<leaf>"])

## [1] 100
```

Count of leaves when the tree is fully grown is 100

CP Table:

CP table is complexity-parameter table of cross-validation errors at respective splits.

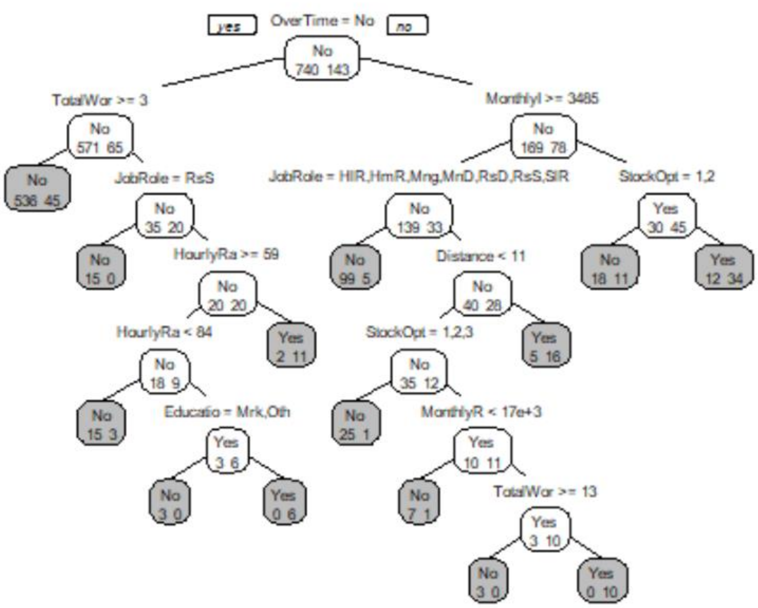
```
hrfulltree$cptable

##             CP nsplit  rel error    xerror       xstd
## 1  0.032846715      0 1.00000000 1.0000000 0.07852059
## 2  0.029197080      2 0.93430657 1.0291971 0.07944449
## 3  0.024330900      7 0.78832117 1.0072993 0.07875373
## 4  0.021897810     10 0.71532847 0.9854015 0.07804990
## 5  0.014598540     16 0.58394161 0.9635036 0.07733262
## 6  0.012165450     21 0.51094891 0.9781022 0.07781232
## 7  0.010948905     24 0.47445255 1.0145985 0.07898542
## 8  0.007299270     28 0.43065693 1.0218978 0.07921567
```

```
## 9  0.006737788     49 0.27737226 1.1313869 0.08250461
## 10 0.005474453     71 0.12408759 1.1313869 0.08250461
## 11 0.004866180     79 0.08029197 1.1605839 0.08333220
## 12 0.003649635     85 0.05109489 1.2043796 0.08453697
## 13 0.000000000     99 0.00000000 1.2481752 0.08569944
```

Pruning Tree:

Pruning of a tree is done using the CP Table, where the tree is constructed with the CP value that has least Cross-Validation error (xerror).

```
hrpruned <- prune(hrfulltree,
                   cp = hrfulltree$cptable[which.min(hrfulltree$cptable[,"xerror"]),"CP"]
)
length(hrpruned$frame$var[hrpruned$frame$var == "<leaf>"])

## [1] 17

prp(hrpruned, type = 1, extra = 1, split.font = 1, varlen = -10,
    box.col=ifelse(hrpruned$frame$var == "<leaf>", 'gray', 'white'))
```



Accuracies of Training and Validation Datasets

```
### Confusion Matrices
### for Training set

hrtrainCM <- predict(hrtree, data = hr.train,type = "class")
confusionMatrix(hrtrainCM, as.factor(hr.train$Attrition))

## Confusion Matrix and Statistics
##
##             Reference
## Prediction   0   1
```

```
##          0 734  90
##          1  11  47
##
##                Accuracy : 0.8855
##                  95% CI : (0.8626, 0.9058)
##     No Information Rate : 0.8447
##     P-Value [Acc > NIR] : 0.0003121
##
##                   Kappa : 0.4293
##  Mcnemar's Test P-Value : 0.000000000000008407
##
##             Sensitivity : 0.9852
##             Specificity : 0.3431
##          Pos Pred Value : 0.8908
##          Neg Pred Value : 0.8103
##              Prevalence : 0.8447
##          Detection Rate : 0.8322
##    Detection Prevalence : 0.9342
##       Balanced Accuracy : 0.6642
##
##        'Positive' Class : 0
##
```

Accuracy in Training Dataset is 88.55%

```
### for Validation set

hrvalidCM <- predict(hrtree,newdata = hr.valid, type = "class")
confusionMatrix(hrvalidCM, as.factor(hr.valid$Attrition))
```

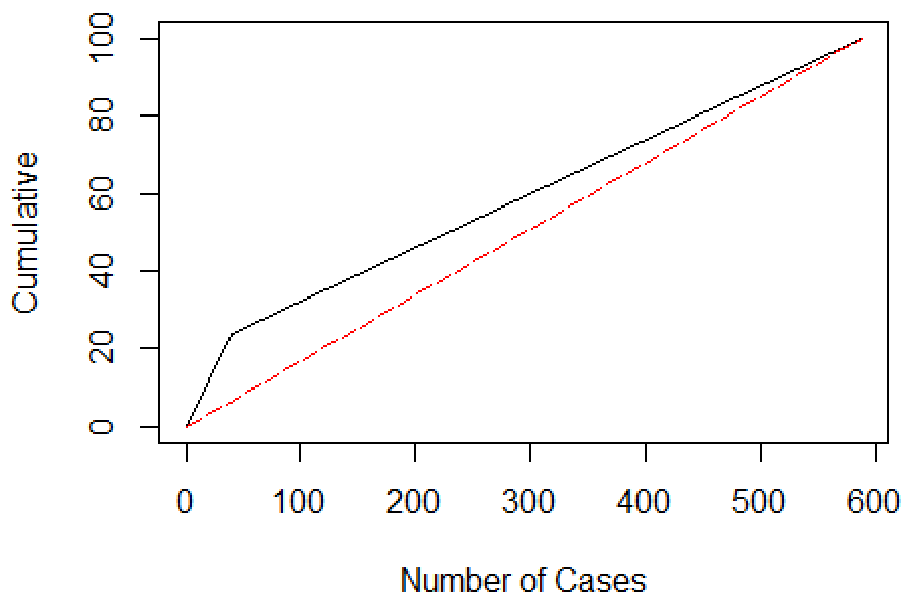
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 473  76
##          1  15  24
##
##                Accuracy : 0.8452
##                  95% CI : (0.8134, 0.8735)
##     No Information Rate : 0.8299
##     P-Value [Acc > NIR] : 0.1758
##
##                   Kappa : 0.2763
##  Mcnemar's Test P-Value : 0.0000000003181
##
##             Sensitivity : 0.9693
##             Specificity : 0.2400
##          Pos Pred Value : 0.8616
##          Neg Pred Value : 0.6154
##              Prevalence : 0.8299
##          Detection Rate : 0.8044
##    Detection Prevalence : 0.9337
##       Balanced Accuracy : 0.6046
##
```

```
##        'Positive' Class : 0
##
```

Accuracy in Validation Dataset is 84.52%

Lift Chart:

```
gain <- gains(as.numeric(hr.valid$Attrition),as.numeric(hrvalidCM)-1, groups = 10)

plot(c(0,gain$cume.pct.of.total*sum(hr.valid$Attrition))~c(0,gain$cume.obs),
     xlab = "Number of Cases", ylab = "Cumulative", main = "", type = "l")
lines(c(0,sum(hr.valid$Attrition))~c(0, dim(hr.valid)[1]), col= "red", lty = 5)
```



Area under the curve:

```
roc_obj <- roc(as.numeric(hr.valid$Attrition), as.numeric(hrvalidCM))
auc(roc_obj)

## Area under the curve: 0.6046
```

Running the CART model using a BALANCED training data set:

Number of 1’s and 0’s in training data set before balancing

```
table(hr.train$Attrition)

##
##   0   1
## 745 137
```

Balancing the dataset using ROSE function.

Rose function is an oversampling technique that deals with imbalanced dataset by creating synthetic data.

```
hr.train <- ROSE(Attrition ~ ., data = hr.train, seed = 1)$data
```

Number of 1's and 0's in training data set after balancing

```
table(hr.train$Attrition)

##
##   0   1
## 455 427
```

Running rpart() using new balanced training data set

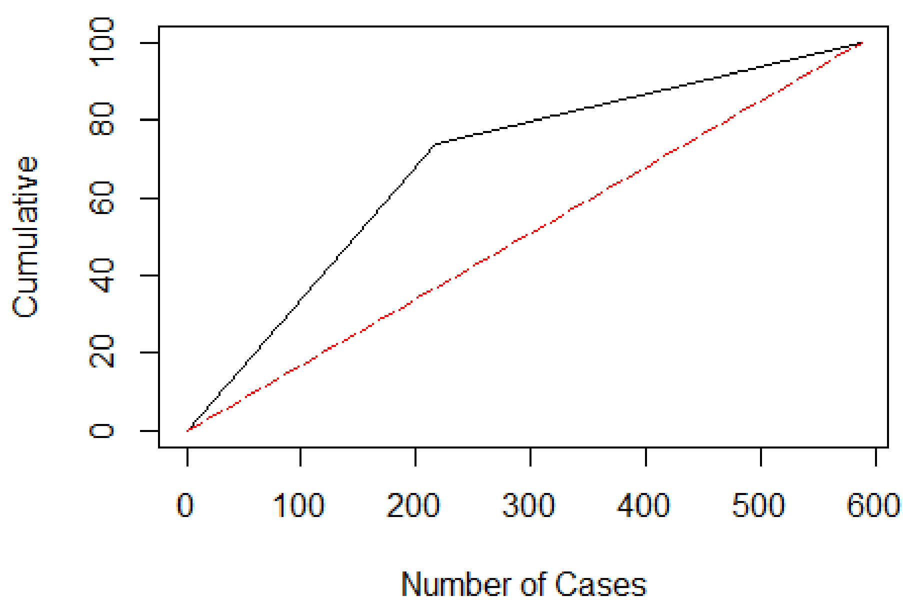
```
# Generate classification tree
hrtree <- rpart( Attrition~ ., data = hr.train, method = "class")

### Fully-Grown Tree
hrfulltree <- rpart(Attrition ~ ., data = hr.train,
                    method = "class", cp = 0, minsplit = 1)

#pruning tree

hrpruned <- prune(hrfulltree,
                  cp = hrfulltree$cptable[which.min(hrfulltree$cptable[,"xerror"]),"CP"])


### Confusion Matrices

### for Validation set

hrvalidCM <- predict(hrtree,newdata = hr.valid, type = "class")
confusionMatrix(hrvalidCM, as.factor(hr.valid$Attrition))

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 345  26
##          1 143  74
##
##                Accuracy : 0.7126
##                  95% CI : (0.6742, 0.7489)
##     No Information Rate : 0.8299
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3051
##  Mcnemar's Test P-Value : <0.0000000000000002
##
##             Sensitivity : 0.7070
##             Specificity : 0.7400
##          Pos Pred Value : 0.9299
##          Neg Pred Value : 0.3410
```

```
##               Prevalence : 0.8299
##           Detection Rate : 0.5867
##     Detection Prevalence : 0.6310
##        Balanced Accuracy : 0.7235
##
##         'Positive' Class : 0
##
```

Validation Accuracy is 71.26%, specificity is 74%.

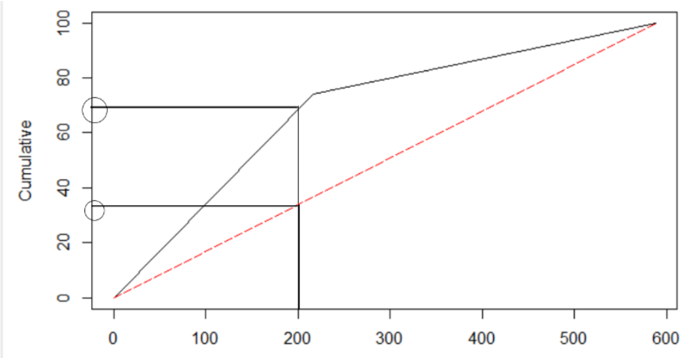
Lift Chart and AUC after balancing:

```
gain <- gains(as.numeric(hr.valid$Attrition),as.numeric(hrvalidCM)-1, groups = 10)

 plot(c(0,gain$cume.pct.of.total*sum(hr.valid$Attrition))~c(0,gain$cume.obs),
     xlab = "Number of Cases", ylab = "Cumulative", main = "", type = "l")
lines(c(0,sum(hr.valid$Attrition))~c(0, dim(hr.valid)[1]), col= "red", lty = 5)
```




```
roc_obj <- roc(as.numeric(hr.valid$Attrition), as.numeric(hrvalidCM))
auc(roc_obj)

## Area under the curve: 0.7235
```

Inference of Lift Chart:



Lift chart is used for profiling. When top 20% (or 200) of records are picked, our model performs 2 times better than the Naives benchmark. (70/30, Point where black line meets Y axis when X = 200 / Point where red dotted line meets Y axis when X=200

Inference of CART Model:

Original Dataset model vs Balanced dataset model, which one to use?

It depends on the business requirement, if a model that yields good accuracy is to be selected then Original Dataset Model which had an accuracy of 84.52% is the right choice. If a model that identifies high number of "Class of Interest" members correctly is to be selected, then the balanced dataset model which identified 74% "Class of Interest" members correctly is the right choice.

KNN: k-Nearest Neighbor

It is a non-parametric method of machine learning algorithm used for classification and regression problem. It aims at classifying records based on similar records in the training data. It is based on distance between records and is data driven where no assumption is made about relationship between Y and X's.

Dataset Preparation

We will be using our initial dataset of attrition provided by IBM to build our model based on K Nearest neighbor algorithm to predict Employee pattern on Attrition using a list of 33 metrics. We will be taking the column which is having numerical data, categorical data is considered only after taking it into numerical form, since it is a KNN algorithm. We have only considered nominal categorical variable into account since converting them into numbers would not distort the representation in the data, whereas converting ordinal variable into numerical would not make sense in KNN.

We will be first reading the dataset into R using 'read.csv' and then defining the required libraries for running the K Nearest neighbor algorithm.

```
#Defining Librarie

library(caret)
library(FNN)
library(gmodels)

##
## Attaching package: 'gmodels'

## The following object is masked from 'package:pROC':
##
##     ci

#Reading preprocessed file for kNN into the system

hr.df<-read.csv("hr_Analytics_knn.csv",stringsAsFactors = FALSE)

names(hr.df)[1]<-"Attrition"

#Initial Exploration of the dataset

str(hr.df)

## 'data.frame':    1470 obs. of  28 variables:
##  $ Attrition                : chr  "Yes" "No" "Yes" "No" ...
##  $ DailyRate                : int  1102 279 1373 1392 591 1005 1324 1358 216 1299 ...
##  $ DistanceFromHome         : int  1 8 2 3 2 2 3 24 23 27 ...
##  $ Education                : int  2 1 2 4 1 2 3 1 3 3 ...
##  $ EmployeeCount            : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ EmployeeNumber           : int  1 2 4 5 7 8 10 11 12 13 ...
##  $ EnvironmentSatisfaction  : int  2 3 4 4 1 4 3 4 4 3 ...
##  $ Gender                   : int  0 1 1 0 1 1 0 1 1 1 ...
##  $ JobInvolvement           : int  3 2 2 3 3 3 4 3 2 3 ...
##  $ JobLevel                 : int  2 2 1 1 1 1 1 1 3 2 ...
##  $ JobSatisfaction          : int  4 2 3 3 2 4 1 3 3 3 ...
##  $ MonthlyRate              : int  19479 24907 2396 23159 16632 11864 9964 13335 8787 1
6577 ...
##  $ NumCompaniesWorked       : int  8 1 6 1 9 0 4 1 0 6 ...
##  $ Over18                   : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ OverTime                : int  1 0 1 1 0 0 1 0 0 0 ...
##  $ 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 ...
##  $ Age                     : int  41 49 37 33 27 32 59 30 38 36 ...
```

```
table(hr.df$Attrition)
```

```
##
##   No  Yes
## 1233  237
```

Training and Validation Dataset

We will be dividing our data into Training and Validation data.

. Training Data: We will be using this data-set to define the set of rules or build a KNN model from the predictor variables to predict the outcome of Employee Attrition

. Validation Data: It is used to test the model accuracy based on data-set which was not used for building the model

We have split the data into 60% training data-set and 40% validation data-set. Below is code for dividing data-set into training and validation data-set:

```
### Partitioning data
set.seed(1)
train.index <- sample(row.names(hr.df), 0.6*dim(hr.df)[1])
valid.index <- setdiff(row.names(hr.df), train.index)
train.df <- hr.df[train.index, ]
valid.df <- hr.df[valid.index, ]
train.df.labels<-hr.df[train.index,1]
valid.df.labels<-hr.df[valid.index,1]
```

Normalization

Before moving on to running a KNN algorithm, the data must be normalized i.e. it should follow a normal distribution. We will be using processed function from CARET to normalize the training and validation data-set.

```
### Run K-NN
train.norm.df <- train.df
valid.norm.df <- valid.df

### Normalize data using preProcess() from CARET
set.seed(111)
norm.values <- preProcess(train.df[, 2:28], method=c("center", "scale"))
```

```
## Warning in preProcess.default(train.df[, 2:28], method = c("center",
## "scale")): These variables have zero variances: EmployeeCount, Over18,
## StandardHours

train.norm.df[, 2:28] <- predict(norm.values, train.df[, 2:28])
valid.norm.df[, 2:28] <- predict(norm.values, valid.df[, 2:28])
valid.norm.df$Attrition<-as.factor(valid.norm.df$Attrition)
```

Build model

We will be first running the model with a random K number.

```
knn.pred <- knn(train.norm.df[, 2:28], valid.norm.df[, 2:28],
                cl = train.norm.df[, 1], k = 7)
```

Choosing Optimal K

We will be choosing the best K value using the below code based on the maximum accuracy provided by the confusion matrix.

```
### Chooose optimal K

### Initialize a data frame with two columns: k and accuracy
accuracy.df <- data.frame(k = seq(1, 14, 1), accuracy = rep(0, 14))

### compute knn for different k on validation
for(i in 1:14) {
  knn.pred <- knn(train.norm.df[, 2:28], valid.norm.df[, 2:28],
                  cl = train.norm.df[, 1], k = i)
  accuracy.df[i, 2] <- confusionMatrix(knn.pred, valid.norm.df[, 1])$overall[1]
}

## Warning in confusionMatrix.default(knn.pred, valid.norm.df[, 1]): Levels
## are not in the same order for reference and data. Refactoring data to
## match.

accuracy.df

##     k  accuracy
## 1   1 0.7908163
## 2   2 0.8197279
## 3   3 0.8265306
## 4   4 0.8299320
## 5   5 0.8333333
## 6   6 0.8316327
## 7   7 0.8333333
## 8   8 0.8316327
## 9   9 0.8299320
## 10 10 0.8333333
## 11 11 0.8316327
## 12 12 0.8350340
## 13 13 0.8350340
## 14 14 0.8299320
```

Model Accuracy

We achieved an overall accuracy of ~84%. Below is the cross table for that:

```
####Running with optimum K

knn.pred <- knn(train.norm.df[, 2:28], valid.norm.df[, 2:28],
                cl = train.norm.df[, 1], k = 5)

CrossTable(x=valid.df.labels, y=knn.pred, prop.chisq=FALSE)

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  588
##
##
##                 | knn.pred
## valid.df.labels |        No |       Yes | Row Total |
## ----------------|-----------|-----------|-----------|
##              No |       483 |         5 |       488 |
##                 |     0.990 |     0.010 |     0.830 |
##                 |     0.839 |     0.417 |           |
##                 |     0.821 |     0.009 |           |
## ----------------|-----------|-----------|-----------|
##             Yes |        93 |         7 |       100 |
##                 |     0.930 |     0.070 |     0.170 |
##                 |     0.161 |     0.583 |           |
##                 |     0.158 |     0.012 |           |
## ----------------|-----------|-----------|-----------|
##    Column Total |       576 |        12 |       588 |
##                 |     0.980 |     0.020 |           |
## ----------------|-----------|-----------|-----------|
##
##
```

Linear Discriminant Analysis

Removing unnecessary columns

```
hr_data$StandardHours<- NULL
hr_data$EmployeeCount<- NULL
hr_data$Over18<- NULL
hr_data$EmployeeNumber<- NULL
```

Data Partition

```
row<- seq(1,nrow(hr_data),1)
set.seed(10)
train_rows<- sample(row, 0.7*nrow(hr_data))
train <- hr_data[train_rows, ]
valid <- hr_data[-train_rows, ]
```

Normalize the data

Estimate preprocessing parameters

```
library(caret)
norm.values  <- preProcess(train, method = c("center", "scale"))
```

Transform the data using the estimated parameters

```
train.norm <- predict(norm.values, train)
valid.norm <- predict(norm.values, valid)
```

run lda()

```
library(MASS)
lda1 <- lda(Attrition~., data = train.norm)
lda1$counts

##  No Yes
## 863 166
```

output

LDA uses means and variances of each class in order to create a linear boundary between them. This boundary is delimited by the coefficients. Prior probabilities of groups: These probabilities are the ones that already exist in your training data. You can see in the output that the probabilities of groups for No is 84.01% and that for Yes is 15.98%. Group means: This gives is the average of each predictor within each class.

```
lda1

## Call:
## lda(Attrition ~ ., data = train.norm)
##
## Prior probabilities of groups:
##        No       Yes
## 0.8386783 0.1613217
##
## Group means:
##          ï..Age BusinessTravelTravel_Frequently
```

```
## No   0.05978277                       0.1703360
## Yes -0.31079839                       0.2771084
##     BusinessTravelTravel_Rarely  DailyRate
## No                    0.7149479  0.0293545
## Yes                   0.6686747 -0.1526080
##     DepartmentResearch & Development DepartmentSales DistanceFromHome
## No                         0.6685979       0.2908459      -0.02300254
## Yes                        0.5783133       0.3795181       0.11958549
##     Education2 Education3 Education4 Education5
## No   0.1900348  0.3626883  0.2954809 0.03592121
## Yes  0.2048193  0.3734940  0.2530120 0.02409639
##     EducationFieldLife Sciences EducationFieldMarketing
## No                    0.4171495              0.09965238
## Yes                   0.3975904              0.12650602
##     EducationFieldMedical EducationFieldOther
## No              0.3290846          0.05214368
## Yes             0.2831325          0.05421687
##     EducationFieldTechnical Degree EnvironmentSatisfaction2
## No                      0.08458864                0.2027810
## Yes                     0.12048193                0.1987952
##     EnvironmentSatisfaction3 EnvironmentSatisfaction4 GenderMale
## No                 0.3198146                0.3128621  0.6060255
## Yes                0.2650602                0.2409639  0.6325301
##       HourlyRate JobInvolvement2 JobInvolvement3 JobInvolvement4 JobLevel2
## No  -0.002140168       0.2363847       0.6071842      0.10892236 0.3939745
## Yes  0.011126296       0.3012048       0.5602410      0.04819277 0.2530120
##     JobLevel3  JobLevel4  JobLevel5 JobRoleHuman Resources
## No  0.1390498 0.07879490 0.05677868             0.03012746
## Yes 0.1325301 0.01204819 0.01204819             0.04216867
##     JobRoleLaboratory Technician JobRoleManager
## No                     0.1714948     0.07995365
## Yes                    0.2590361     0.01807229
##     JobRoleManufacturing Director JobRoleResearch Director
## No                     0.09733488              0.054461182
## Yes                    0.04819277              0.006024096
##     JobRoleResearch Scientist JobRoleSales Executive
## No                  0.2039397              0.2178447
## Yes                 0.2108434              0.2530120
##     JobRoleSales Representative JobSatisfaction2 JobSatisfaction3
## No                   0.04403244        0.1900348        0.3024334
## Yes                  0.12048193        0.2289157        0.3072289
##     JobSatisfaction4 MaritalStatusMarried MaritalStatusSingle
## No         0.3244496            0.4831981           0.2804171
## Yes        0.1867470            0.3373494           0.5301205
##     MonthlyIncome  MonthlyRate NumCompaniesWorked OverTimeYes
## No     0.07419313  0.002217477        -0.02492306   0.2410197
## Yes   -0.38571491 -0.011528208         0.12956988   0.5361446
##     PercentSalaryHike PerformanceRating4 RelationshipSatisfaction2
## No      -0.0006006743          0.1483198                 0.2201622
## Yes      0.0031227824          0.1566265                 0.2048193
##     RelationshipSatisfaction3 RelationshipSatisfaction4 StockOptionLevel1
## No                  0.3024334                 0.3035921         0.4380070
## Yes                 0.2951807                 0.2530120         0.2048193
##     StockOptionLevel2 StockOptionLevel3 TotalWorkingYears
## No         0.11008111        0.06025492        0.08104103
```

```
## Yes        0.05421687        0.06024096       -0.42131570
##     TrainingTimesLastYear WorkLifeBalance2 WorkLifeBalance3
## No            0.03808181        0.2294322        0.6338355
## Yes          -0.19797953        0.2108434        0.5542169
##     WorkLifeBalance4 YearsAtCompany YearsInCurrentRole
## No        0.09385863     0.06631627         0.07590849
## Yes       0.12048193    -0.34476470        -0.39463267
##     YearsSinceLastPromotion YearsWithCurrManager
## No               0.01491823           0.07168786
## Yes             -0.07755681          -0.37269050
##
## Coefficients of linear discriminants:
##                                          LD1
## ï..Age                           -0.06117341
## BusinessTravelTravel_Frequently   0.93220846
## BusinessTravelTravel_Rarely       0.46335040
## DailyRate                        -0.10207411
## DepartmentResearch & Development  0.04890989
## DepartmentSales                  -0.03058621
## DistanceFromHome                  0.15353036
## Education2                        0.16171439
## Education3                        0.01922632
## Education4                       -0.02243733
## Education5                        0.01629280
## EducationFieldLife Sciences      -0.29705424
## EducationFieldMarketing          -0.18897506
## EducationFieldMedical            -0.33258497
## EducationFieldOther              -0.21384494
## EducationFieldTechnical Degree    0.20782262
## EnvironmentSatisfaction2         -0.56955407
## EnvironmentSatisfaction3         -0.65517073
## EnvironmentSatisfaction4         -0.72112746
## GenderMale                        0.16370898
## HourlyRate                        0.01521711
## JobInvolvement2                  -0.58707874
## JobInvolvement3                  -0.75711979
## JobInvolvement4                  -1.15244734
## JobLevel2                        -0.77135073
## JobLevel3                        -0.11520397
## JobLevel4                        -0.17472931
## JobLevel5                         0.21339369
## JobRoleHuman Resources            0.08756491
## JobRoleLaboratory Technician      0.25000141
## JobRoleManager                    0.24397646
## JobRoleManufacturing Director     0.28345123
## JobRoleResearch Director         -0.15281015
## JobRoleResearch Scientist        -0.32277816
## JobRoleSales Executive            0.75865402
## JobRoleSales Representative       0.66321162
## JobSatisfaction2                 -0.27166365
## JobSatisfaction3                 -0.37668054
## JobSatisfaction4                 -0.78942687
## MaritalStatusMarried             -0.04199850
## MaritalStatusSingle               0.20299200
## MonthlyIncome                    -0.24010192
```

```
## MonthlyRate                        -0.01072499
## NumCompaniesWorked                  0.26847439
## OverTimeYes                         1.23008642
## PercentSalaryHike                   0.03744176
## PerformanceRating4                  0.01205462
## RelationshipSatisfaction2          -0.39196052
## RelationshipSatisfaction3          -0.42287115
## RelationshipSatisfaction4          -0.55413102
## StockOptionLevel1                  -0.74205555
## StockOptionLevel2                  -0.55485235
## StockOptionLevel3                  -0.40718416
## TotalWorkingYears                  -0.22459542
## TrainingTimesLastYear              -0.13735373
## WorkLifeBalance2                   -0.86347944
## WorkLifeBalance3                   -1.05068074
## WorkLifeBalance4                   -0.65931561
## YearsAtCompany                      0.13007811
## YearsInCurrentRole                 -0.17862043
## YearsSinceLastPromotion             0.18552159
## YearsWithCurrManager               -0.14883942
```

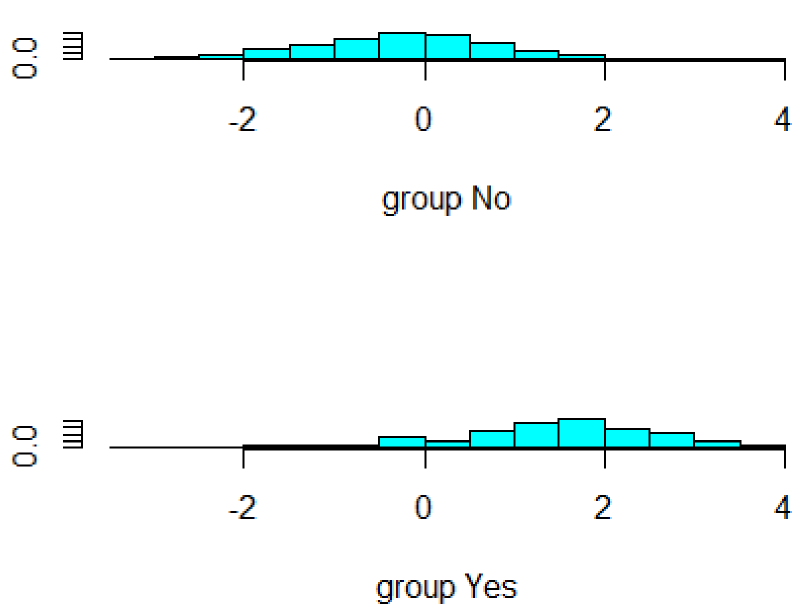
```
prop.ld1 = lda1$svd^2/sum(lda1$svd^2)
prop.ld1
```

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

predict - using training data and plot

We can see that there is lot of overlapping between Yes and No.

```
pred1.train <- predict(lda1, train.norm)
ldahist(data = pred1.train$x[,1], g = train.norm$Attrition)
```



Predict - using validation data

```
pred2.valid <- predict(lda1, valid.norm)
names(pred2.valid)

## [1] "class"     "posterior" "x"
```

Model accuracy

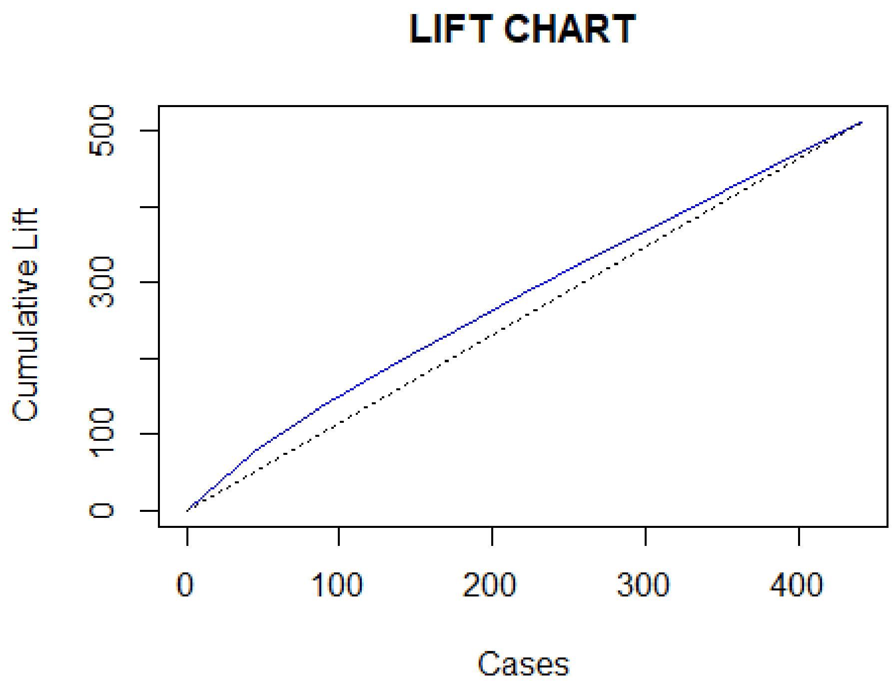
```
table(pred2.valid$class, valid.norm$Attrition)

##
##         No Yes
##    No  362  39
##    Yes   8  32

mean(pred2.valid$class == valid.norm$Attrition)

## [1] 0.893424

sum(pred2.valid$posterior[, 1] >=.5)

## [1] 401

sum(pred2.valid$posterior[, 1] >=.75)

## [1] 350
```

lift chart

```
library(gains)
gain <- gains(as.numeric(valid.norm$Attrition), pred2.valid$x[,1], groups = 10)
```

Gains

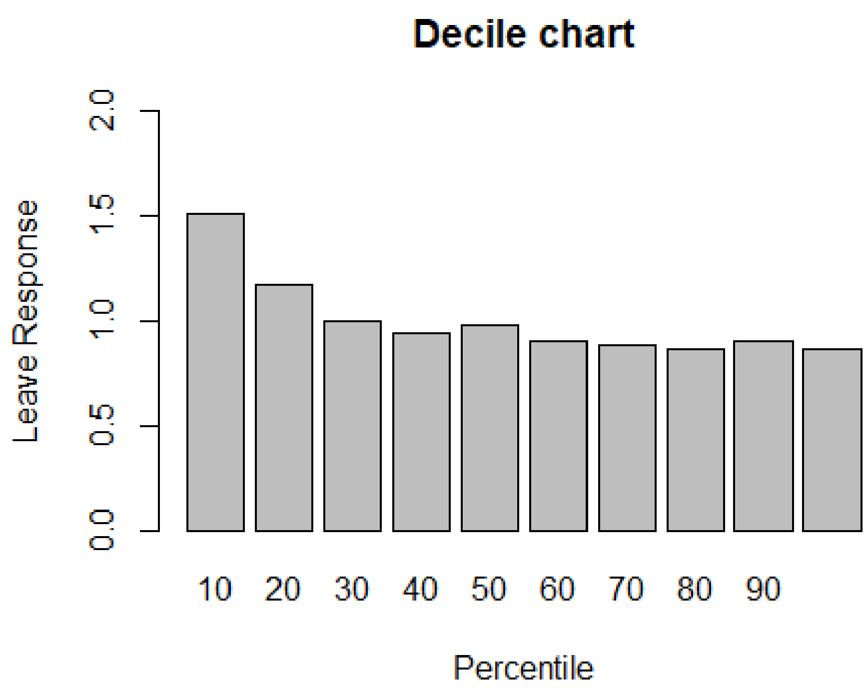
```
valid.norm$Attrition<- as.numeric(valid.norm$Attrition)
plot(c(0,gain$cume.pct.of.total*sum(valid.norm$Attrition))~c(0,gain$cume.obs),
     xlab="Cases", ylab="Cumulative Lift", main="LIFT CHART",
     col = "blue1", type="l")
lines(c(0,sum(valid.norm$Attrition))~c(0, dim(valid)[1]), lty = 9)
```



Plot decile-wise chart

You can see that all the deciles are in the descending order which is a good sign of decile chart.The records are sorted by their predicted scores. The top decile contains the 10% of the employees most likely with Yes and the bottom decile contains the 10% of the employees. Also, it tells us that out LDA model preforms better for top 20% deciles compared to the naive model.

```
heights <- gain$mean.resp/mean(valid.norm$Attrition)
barplot(heights, names.arg = gain$depth,  ylim = c(0,2),
                 xlab = "Percentile", ylab = "Leave Response",
                 main = "Decile chart")
```



Conclusion

Which Model is best?

Table of validation dataset accuracies:

	Validation Accuracy
KNN	84
CART	84.52
LDA	89.34
Logistic	86.2

As shown in the table above, LDA is the best model as it has an accuracy of 89.34%.

Why?

When the data is normally distributed LDA yields best accuracy and it can outperform any classification model, so most of the columns that are used as predictors to run the model in the dataset must have been normally distributed, which is why LDA outperformed all the other classification models. Also, when data is normally distributed LDA can work efficiently even with a small dataset (with as less as 20 observations), since the dataset we used is relatively small, LDA outperformed all the other classification models.

REFRENCES AND CITATIONS

- https://towardsdatascience.com/people-analytics-with-attrition-predictions-12adcce9573f
- https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset
- https://sebastianraschka.com/Articles/2014_python_lda.html
- https://www.thebalancecareers.com/top-reasons-why-employees-quit-their-job-1918985
- https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/
- https://www.statisticssolutions.com/what-is-logistic-regression/
- https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7
- https://www.saedsayad.com/model_evaluation.htm