**HR Employee Attrition**

**Data Import (Target variable is "Attrition" column)**

Read the file in R

cart <- read.table("HR\_Employee\_Attrition\_Data.csv",sep = ",", header = T)

**Perform Exploratory Data Analysis**

str(cart) # DATA TYPE

'data.frame': 2940 obs. of 31 variables:

$ Attrition : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 1 1 ...

$ BusinessTravel : Factor w/ 3 levels "Non-Travel","Travel\_Frequently",..: 3 3 2 3 3 1 2 3 3 3 ...

$ Department : Factor w/ 3 levels "Human Resources",..: 2 2 2 2 2 2 2 3 2 2 ...

$ Education : int 3 4 4 3 4 2 4 3 4 3 ...

$ EducationField : Factor w/ 6 levels "Human Resources",..: 5 2 4 4 4 2 2 4 5 4 ...

$ EnvironmentSatisfaction : int 3 2 3 1 4 3 3 4 3 4 ...

$ JobInvolvement : int 3 3 3 2 3 2 2 2 3 3 ...

$ JobLevel : int 2 2 2 2 5 2 2 3 1 2 ...

$ JobRole : Factor w/ 9 levels "Healthcare Representative",..: 5 5 1 7 6 1 3 8 7 1 ...

$ JobSatisfaction : int 1 2 2 3 4 4 3 1 1 2 ...

$ MaritalStatus : Factor w/ 3 levels "Divorced","Married",..: 2 2 2 3 1 1 1 2 2 3 ...

$ OverTime : Factor w/ 2 levels "No","Yes": 1 1 2 2 1 1 1 1 1 1 ...

$ PercentSalaryHike : int 15 12 14 14 17 14 11 17 12 12 ...

$ RelationshipSatisfaction: int 3 2 4 1 4 4 2 3 4 4 ...

$ StockOptionLevel : int 1 1 2 0 2 1 1 0 1 0 ...

$ TrainingTimesLastYear : int 1 2 3 3 0 2 3 3 3 2 ...

$ WorkLifeBalance : int 2 2 3 1 3 4 4 3 2 2 ...

$ YearsAtCompany : int 6 6 8 20 2 1 9 13 8 5 ...

$ YearsInCurrentRole : int 2 5 7 7 2 0 7 12 2 2 ...

$ YearsSinceLastPromotion : int 0 4 6 1 2 0 0 6 7 2 ...

$ stability : num 1 1 3 5 2 2 3 1 1 1 ...

$ AgeGroup : num 2 4 7 5 6 4 5 3 6 5 ...

$ DistanceGroup : num 4 1 4 2 1 1 2 2 3 1 ...

$ YearsWithManagerGroup : num 1 2 3 3 1 1 1 1 3 2 ...

$ WorkYearGroup : num 3 3 5 5 6 5 3 4 3 3 ...

$ NumCompGroup : num 1 3 2 1 3 3 1 3 3 3 ...

$ DailyRateGroup : num 5 2 1 4 2 5 3 3 5 3 ...

$ HourlyRateGroup : num 6 4 4 5 1 7 6 3 4 2 ...

$ MonthlyRateGroup : num 2 4 3 5 5 4 2 5 4 2 ...

$ MonthlyIncomeGroup : num 4 3 3 2 7 2 2 4 1 2 ...

*Brief summary for data set*

summary(cart) # Employee count, Over18 have just one value. This could be removed

Attrition BusinessTravel Department Education

No :2466 Non-Travel : 300 Human Resources : 126 Min. :1.000000

Yes: 474 Travel\_Frequently: 554 Research & Development:1922 1st Qu.:2.000000

Travel\_Rarely :2086 Sales : 892 Median :3.000000

Mean :2.912925

3rd Qu.:4.000000

Max. :5.000000

EducationField EnvironmentSatisfaction JobInvolvement JobLevel

Human Resources : 54 Min. :1.000000 Min. :1.000000 Min. :1.000000

Life Sciences :1212 1st Qu.:2.000000 1st Qu.:2.000000 1st Qu.:1.000000

Marketing : 318 Median :3.000000 Median :3.000000 Median :2.000000

Medical : 928 Mean :2.721769 Mean :2.729932 Mean :2.063946

Other : 164 3rd Qu.:4.000000 3rd Qu.:3.000000 3rd Qu.:3.000000

Technical Degree: 264 Max. :4.000000 Max. :4.000000 Max. :5.000000

JobRole JobSatisfaction MaritalStatus OverTime PercentSalaryHike

Sales Executive :652 Min. :1.000000 Divorced: 654 No :2108 Min. :11.00000

Research Scientist :584 1st Qu.:2.000000 Married :1346 Yes: 832 1st Qu.:12.00000

Laboratory Technician :518 Median :3.000000 Single : 940 Median :14.00000

Manufacturing Director :290 Mean :2.728571 Mean :15.20952

Healthcare Representative:262 3rd Qu.:4.000000 3rd Qu.:18.00000

Manager :204 Max. :4.000000 Max. :25.00000

(Other) :430

RelationshipSatisfaction StockOptionLevel TrainingTimesLastYear WorkLifeBalance

Min. :1.000000 Min. :0.0000000 Min. :0.00000 Min. :1.000000

1st Qu.:2.000000 1st Qu.:0.0000000 1st Qu.:2.00000 1st Qu.:2.000000

Median :3.000000 Median :1.0000000 Median :3.00000 Median :3.000000

Mean :2.712245 Mean :0.7938776 Mean :2.79932 Mean :2.761224

3rd Qu.:4.000000 3rd Qu.:1.0000000 3rd Qu.:3.00000 3rd Qu.:3.000000

Max. :4.000000 Max. :3.0000000 Max. :6.00000 Max. :4.000000

YearsAtCompany YearsInCurrentRole YearsSinceLastPromotion stability

Min. : 0.000000 Min. : 0.000000 Min. : 0.000000 Min. :1.000000

1st Qu.: 3.000000 1st Qu.: 2.000000 1st Qu.: 0.000000 1st Qu.:1.000000

Median : 5.000000 Median : 3.000000 Median : 1.000000 Median :2.000000

Mean : 7.008163 Mean : 4.229252 Mean : 2.187755 Mean :2.140816

3rd Qu.: 9.000000 3rd Qu.: 7.000000 3rd Qu.: 3.000000 3rd Qu.:3.000000

Max. :40.000000 Max. :18.000000 Max. :15.000000 Max. :9.000000

AgeGroup DistanceGroup YearsWithManagerGroup WorkYearGroup NumCompGroup

Min. :1.000000 Min. :1.000000 Min. :1.000000 Min. :1.000000 Min. :1.000000

1st Qu.:2.000000 1st Qu.:1.000000 1st Qu.:1.000000 1st Qu.:3.000000 1st Qu.:1.000000

Median :4.000000 Median :2.000000 Median :2.000000 Median :3.000000 Median :1.000000

Mean :3.789116 Mean :2.293878 Mean :1.942177 Mean :3.563265 Mean :1.621769

3rd Qu.:5.000000 3rd Qu.:3.000000 3rd Qu.:3.000000 3rd Qu.:4.000000 3rd Qu.:2.000000

Max. :8.000000 Max. :6.000000 Max. :5.000000 Max. :9.000000 Max. :3.000000

DailyRateGroup HourlyRateGroup MonthlyRateGroup MonthlyIncomeGroup

Min. :1.000000 Min. :1.000000 Min. :1.000000 Min. :1.000000

1st Qu.:2.000000 1st Qu.:2.000000 1st Qu.:2.000000 1st Qu.:1.000000

Median :4.000000 Median :4.000000 Median :3.000000 Median :2.000000

Mean :3.682313 Mean :4.067347 Mean :3.361905 Mean :2.606122

3rd Qu.:5.000000 3rd Qu.:6.000000 3rd Qu.:5.000000 3rd Qu.:3.000000

Max. :6.000000 Max. :7.000000 Max. :6.000000 Max. :7.000000

*Check for missing values*

any(is.na(cart))

[1] FALSE

Check baseline attrition rate

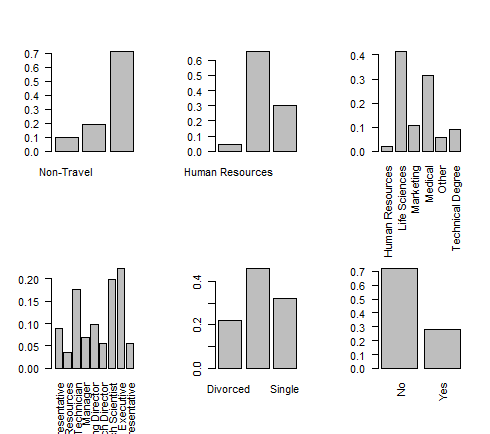
length(which(cart$Attrition=="Yes"))/nrow(cart) # attrition rate = 16%

[1] 0.1612244898

attach(cart)

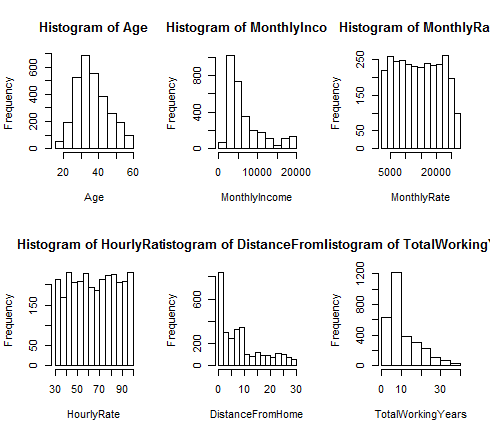
library(ggplot2)

Visual Exploration



Above plot shows the distribution of data across Business travel, Department, Education field, Job Role, Marital Status and Overtime.

If we go deep we could see in each variable there is dominance of certain cateegory like Travel rarely in Business Travel, R&D in Department, Married, No Overtime etc.

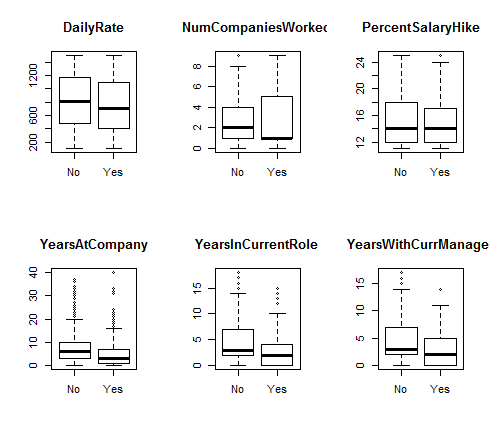


We could observe that majority of the employees are in age of 25-40, with data skewed towards the lower end, suggesting many of them are in starting phase of their career, living closer to the workplace.

Work year group supports the fact that most of them are of 2-3 years of experience

Monthly and hourly rate is more of uniform

Compare the population with attrition factor

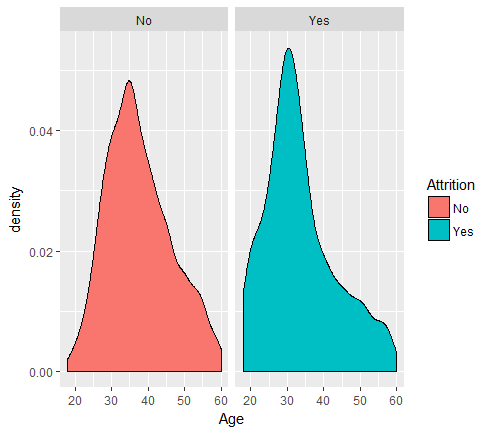


We could see that Years in company, role and manager do influence the attrition factor. Lesser values have more attrition rate then higher value. Basically whenever there is role change (possibly leading to new manager), or new entrants have higher probability to attrite

Apart from that number of companies worked for previously also influence the attrition

Daily rate and Percentage salary hike does not seems that important

ggplot(cart,aes(Age,fill=Attrition))+geom\_density()+facet\_grid(~Attrition)



Observation - People of around age 30-35 have more attrition

prop.table(table(BusinessTravel,Attrition),1)\*100

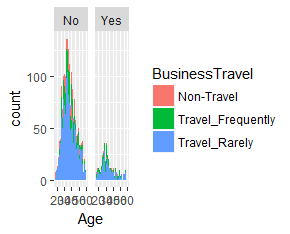
Attrition

BusinessTravel No Yes

Non-Travel 92.00000000 8.00000000

Travel\_Frequently 75.09025271 24.90974729

Travel\_Rarely 85.04314477 14.95685523



# High percentage of frequently travelers in attrition

table(Attrition, Education)

Education

Attrition 1 2 3 4 5

No 278 476 946 680 86

Yes 62 88 198 116 10

prop.table(table(Education,Attrition),1)\*100

Attrition

Education No Yes

1 81.76470588 18.23529412

2 84.39716312 15.60283688

3 82.69230769 17.30769231

4 85.42713568 14.57286432

5 89.58333333 10.41666667

#slight variation in attrition rates for education level 1 and 3

table(Attrition, StockOptionLevel)

StockOptionLevel

Attrition 0 1 2 3

No 954 1080 292 140

Yes 308 112 24 30

prop.table(table(StockOptionLevel,Attrition),1)\*100

Attrition

StockOptionLevel No Yes

0 75.594294770 24.405705230

1 90.604026846 9.395973154

2 92.405063291 7.594936709

3 82.352941176 17.647058824

# High percentage of 0 and 3 in attrition level

**Write Hypothesis and validate the Hypothesis**

1. *Attrition is not dependent on departments*

chisq.test(Attrition,Department)

Pearson's Chi-squared test

data: Attrition and Department

X-squared = 21.592015, df = 2, p-value = 0.00002048111

# significant p value

prop.table(table(Department,Attrition),1)\*100

Attrition

Department No Yes

Human Resources 80.95238095 19.04761905

Research & Development 86.16024974 13.83975026

Sales 79.37219731 20.62780269

# Research has low value, rest two are same

1. Distance does not influence attrition

summary(glm(Attrition~DistanceFromHome, data = cart, family = "binomial"))

Call:

glm(formula = Attrition ~ DistanceFromHome, family = "binomial",

data = cart)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.7341513 -0.5946656 -0.5617801 -0.5365907 2.0046473

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -1.890050774 0.078758876 -23.99794 < 0.000000000000000222 \*\*\*

DistanceFromHome 0.024710074 0.005877666 4.20406 0.000026217 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2597.1654 on 2939 degrees of freedom

Residual deviance: 2580.0052 on 2938 degrees of freedom

AIC: 2584.0052

Number of Fisher Scoring iterations: 4

#Significant p-value hence we reject the null hypothesis

1. Environment satisfaction is not factor for attrition

summary(glm(Attrition~EnvironmentSatisfaction, data = cart, family = "binomial"))

Call:

glm(formula = Attrition ~ EnvironmentSatisfaction, family = "binomial",

data = cart)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.7131845 -0.6368618 -0.5672447 -0.5041556 2.0619205

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -0.98621087 0.12506695 -7.88546 0.0000000000000031337 \*\*\*

EnvironmentSatisfaction -0.25311519 0.04547678 -5.56581 0.0000000260935483544 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2597.1654 on 2939 degrees of freedom

Residual deviance: 2566.1610 on 2938 degrees of freedom

AIC: 2570.161

Number of Fisher Scoring iterations: 4

#Significant p-value hence we reject the null hypothesis.

prop.table(table(EnvironmentSatisfaction,Attrition),1)\*100

Attrition

EnvironmentSatisfaction No Yes

1 74.64788732 25.35211268

2 85.01742160 14.98257840

3 86.31346578 13.68653422

4 86.54708520 13.45291480

# satisfcation level 1 seems to have high attrition

1. No relationship with gender

chisq.test(Attrition,Gender)

Pearson's Chi-squared test with Yates' continuity correction

data: Attrition and Gender

X-squared = 2.3895635, df = 1, p-value = 0.1221477

> # not a significant variable

prop.table(table(Gender,Attrition),1)\*100

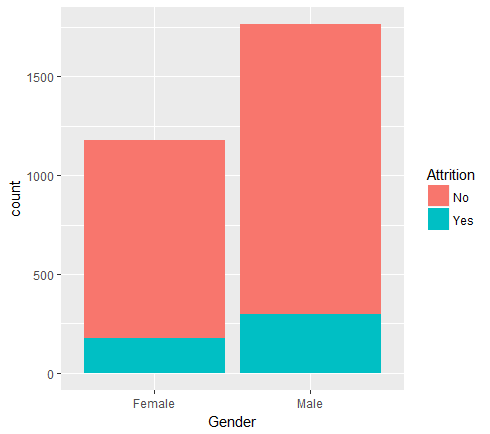
Attrition

Gender No Yes

Female 85.20408163 14.79591837

Male 82.99319728 17.00680272

ggplot(cart,aes(Gender,fill=Attrition))+geom\_bar()



We could observe clearly more males in population

1. Joblevel does not influence attrition

summary(glm(Attrition~JobLevel, data = cart, family = "binomial"))

Call:

glm(formula = Attrition ~ JobLevel, family = "binomial", data = cart)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.7283386 -0.7283386 -0.5738268 -0.3472098 2.5859004

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -0.66259944 0.11254948 -5.88718 0.0000000039283 \*\*\*

JobLevel -0.52897741 0.05934047 -8.91428 < 0.000000000000000222 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2597.1654 on 2939 degrees of freedom

Residual deviance: 2499.7832 on 2938 degrees of freedom

AIC: 2503.7832

Number of Fisher Scoring iterations: 5

#Significant p-value hence we reject the null hypothesis.

prop.table(table(JobLevel,Attrition),1)\*100

Attrition

JobLevel No Yes

1 73.664825046 26.335174954

2 90.262172285 9.737827715

3 85.321100917 14.678899083

4 95.283018868 4.716981132

5 92.753623188 7.246376812

# This seems to have clear demarcation for different levels with job level 1 having highest attrition

1. Joblevel and JobInvolvement not correlated

chisq.test(JobLevel,JobInvolvement)

Pearson's Chi-squared test

data: JobLevel and JobInvolvement

X-squared = 22.152911, df = 12, p-value = 0.03584136

#significantp-value thus these two might have some correlation among them

1. Job Satisfcation not related with attrition

summary(glm(Attrition~JobSatisfaction, data = cart, family = "binomial"))

Call:

glm(formula = Attrition ~ JobSatisfaction, family = "binomial",

data = cart)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.7125536 -0.6369006 -0.5678406 -0.5052012 2.0600353

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -0.99035416 0.12427210 -7.96924 0.0000000000000015965 \*\*\*

JobSatisfaction -0.25097613 0.04504563 -5.57160 0.0000000252413543739 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2597.1654 on 2939 degrees of freedom

Residual deviance: 2566.1083 on 2938 degrees of freedom

AIC: 2570.1083

Number of Fisher Scoring iterations: 4

#Significant p-value hence we reject the null hypothesis.

prop.table(table(JobSatisfaction,Attrition),1)\*100

Attrition

JobSatisfaction No Yes

1 77.16262976 22.83737024

2 83.57142857 16.42857143

3 83.48416290 16.51583710

4 88.67102397 11.32897603

> Satisfaction level 1 has highest attrition which is per our intuition, unsatisfies employees switch early. Rest have approx same level

1. Marital status not influence attrition

chisq.test(Attrition,MaritalStatus)

Pearson's Chi-squared test

data: Attrition and MaritalStatus

X-squared = 92.327353, df = 2, p-value < 0.00000000000000022204

# significant p value

prop.table(table(MaritalStatus,Attrition),1)\*100

Attrition

MaritalStatus No Yes

Divorced 89.90825688 10.09174312

Married 87.51857355 12.48142645

Single 74.46808511 25.53191489

Single population shows greatest attrition rate, again this should be at lower experience level

9) Monthly income not a factor for attrition

summary(glm(Attrition~MonthlyIncome, data = cart, family = "binomial"))

Call:

glm(formula = Attrition ~ MonthlyIncome, family = "binomial",

data = cart)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.7704323 -0.6645891 -0.5811199 -0.3428306 2.6398569

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -0.9291087486 0.0913597088 -10.16979 < 0.000000000000000222 \*\*\*

MonthlyIncome -0.0001271042 0.0000152868 -8.31464 < 0.000000000000000222 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2597.1654 on 2939 degrees of freedom

Residual deviance: 2506.1918 on 2938 degrees of freedom

AIC: 2510.1918

Number of Fisher Scoring iterations: 5

#Significant p-value hence we reject the null hypothesis.

1. Monthly rate and Income not related

cor(MonthlyRate, MonthlyIncome)

[1] 0.03481362613

# very low correlation

1. Num companies does not influence attrition

summary(glm(Attrition~NumCompaniesWorked, data = cart, family = "binomial"))

Call:

glm(formula = Attrition ~ NumCompaniesWorked, family = "binomial",

data = cart)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.6742339 -0.5955840 -0.5711019 -0.5591812 1.9661452

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -1.77652160 0.07520841 -23.62131 < 0.0000000000000002 \*\*\*

NumCompaniesWorked 0.04564641 0.01938825 2.35433 0.018556 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2597.1654 on 2939 degrees of freedom

Residual deviance: 2591.7425 on 2938 degrees of freedom

AIC: 2595.7425

Number of Fisher Scoring iterations: 4

> #Significant p-value hence we reject the null hypothesis.

> prop.table(table(NumCompaniesWorked,Attrition),1)\*100

Attrition

NumCompaniesWorked No Yes

0 88.32487310 11.67512690

1 81.19001919 18.80998081

2 89.04109589 10.95890411

3 89.93710692 10.06289308

4 87.76978417 12.23021583

5 74.60317460 25.39682540

6 77.14285714 22.85714286

7 77.02702703 22.97297297

8 87.75510204 12.24489796

9 76.92307692 23.07692308

We could see that percentage divide in around 2 groups, lees then 5 and greater then 5. Percentage of attrition is fairly same in these two groups

1. Overtime does not influence attrition

summary(glm(Attrition~OverTime, data = cart, family = "binomial"))

Call:

glm(formula = Attrition ~ OverTime, family = "binomial", data = cart)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.8535322 -0.4695137 -0.4695137 -0.4695137 2.1259668

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -2.14964579 0.07123616 -30.17633 < 0.000000000000000222 \*\*\*

OverTimeYes 1.32740619 0.10364216 12.80759 < 0.000000000000000222 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2597.1654 on 2939 degrees of freedom

Residual deviance: 2434.3622 on 2938 degrees of freedom

AIC: 2438.3622

Number of Fisher Scoring iterations: 4

> #Significant p-value hence we reject the null hypothesis.

prop.table(table(OverTime,Attrition),1)\*100

Attrition

OverTime No Yes

No 89.56356736 10.43643264

Yes 69.47115385 30.52884615

> Clear demarcation between the two groups

1. Performance Rating does not influence attrition

summary(glm(Attrition~PerformanceRating, data = cart, family = "binomial"))

Call:

glm(formula = Attrition ~ PerformanceRating, family = "binomial",

data = cart)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.5979766 -0.5920684 -0.5920684 -0.5920684 1.9119466

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -1.7175023 0.4394236 -3.90853 0.000092858 \*\*\*

PerformanceRating 0.0216683 0.1383396 0.15663 0.87554

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2597.1654 on 2939 degrees of freedom

Residual deviance: 2597.1410 on 2938 degrees of freedom

AIC: 2601.141

Number of Fisher Scoring iterations: 3

#Not a significant factor

prop.table(table(PerformanceRating,Attrition),1)\*100

Attrition

PerformanceRating No Yes

3 83.92282958 16.07717042

4 83.62831858 16.37168142

> # Approximately same value, can be ignored

1. Relationship status does not influence attrition

summary(glm(Attrition~RelationshipSatisfaction, data = cart, family = "binomial"))

Call:

glm(formula = Attrition ~ RelationshipSatisfaction, family = "binomial",

data = cart)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.6466573 -0.6140010 -0.5827052 -0.5527526 1.9769919

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -1.3443897 0.1306406 -10.29075 < 0.0000000000000002 \*\*\*

RelationshipSatisfaction -0.1142728 0.0460070 -2.48381 0.012998 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2597.1654 on 2939 degrees of freedom

Residual deviance: 2591.0225 on 2938 degrees of freedom

AIC: 2595.0225

Number of Fisher Scoring iterations: 4

> #Significant p-value hence we reject the null hypothesis.

prop.table(table(RelationshipSatisfaction,Attrition),1)\*100

Attrition

RelationshipSatisfaction No Yes

1 79.34782609 20.65217391

2 85.14851485 14.85148515

3 84.53159041 15.46840959

4 85.18518519 14.81481481

Relationship 1 shows little higher attrition rate

1. Stock Options Level does not influence attrition

summary(glm(Attrition~StockOptionLevel, data = cart, family = "binomial"))

Call:

glm(formula = Attrition ~ StockOptionLevel, family = "binomial",

data = cart)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.6944153 -0.6944153 -0.5485644 -0.4296511 2.4126843

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -1.29954311 0.06456650 -20.12720 < 0.000000000000000222 \*\*\*

StockOptionLevel -0.51833136 0.07086457 -7.31439 0.00000000000025855 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2597.1654 on 2939 degrees of freedom

Residual deviance: 2535.7538 on 2938 degrees of freedom

AIC: 2539.7538

Number of Fisher Scoring iterations: 5

> #Significant p-value hence we reject the null hypothesis.

> prop.table(table(StockOptionLevel,Attrition),1)\*100

Attrition

StockOptionLevel No Yes

0 75.594294770 24.405705230

1 90.604026846 9.395973154

2 92.405063291 7.594936709

3 82.352941176 17.647058824

> Level 0 and 3 have higher attrition rates

**Try with new Features**

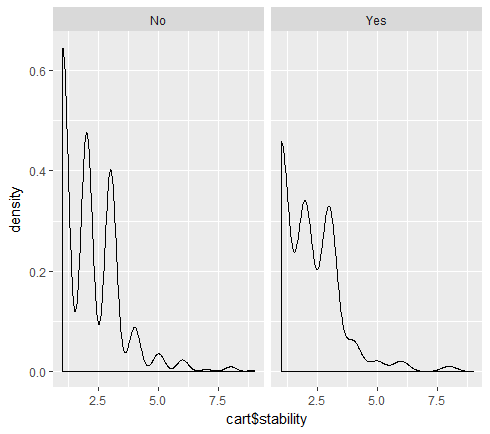
Feature1:

# Pattern of stay in past companies (how stable were they in previous companies).

Assumption here is if people have stayed for longer period then they would also stay approximately same amount of time in present company also. So if ratio is around 1 or greater, they have high chance of leaving

cart$stability = ifelse(cart$NumCompaniesWorked !=0, cart$TotalWorkingYears/cart$NumCompaniesWorked,0)

ggplot(cart,aes(cart$stability))+geom\_density()+facet\_grid(~cart$Attrition)



summary(glm(Attrition~cart$stability, data = cart, family = "binomial"))

Call:

glm(formula = Attrition ~ cart$stability, family = "binomial",

data = cart)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.7078258 -0.7078258 -0.5871382 -0.3973363 3.0269699

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -0.84207080 0.10779324 -7.81191 0.0000000000000056329 \*\*\*

cart$stability -0.41432310 0.05334121 -7.76741 0.0000000000000080107 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2597.1654 on 2939 degrees of freedom

Residual deviance: 2524.1888 on 2938 degrees of freedom

AIC: 2528.1888

Number of Fisher Scoring iterations: 5

# Slight difference is observed visually, but glm shows this is important variable

Feature2:

# Check if salary is comparable to median salary for department and experience level

Assumption is if employee is paid less then median salary for that department and years of experience, he will have more tendancy to switch to better paying job

medianSal = aggregate(MonthlyIncome~Department+TotalWorkingYears, FUN = median)

temp = merge(cart,medianSal, by.x = c("Department","TotalWorkingYears"), by.y = c("Department","TotalWorkingYears"))

cart$CompareSal = temp$MonthlyIncome.x/temp$MonthlyIncome.y

summary(glm(Attrition~cart$CompareSal, data = cart, family = "binomial"))

Call:

glm(formula = Attrition ~ cart$CompareSal, family = "binomial",

data = cart)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.6961580 -0.5963052 -0.5873100 -0.5736685 1.9627787

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -1.8287521 0.1357694 -13.46955 < 0.0000000000000002 \*\*\*

cart$CompareSal 0.1669928 0.1162182 1.43689 0.15075

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2597.1654 on 2939 degrees of freedom

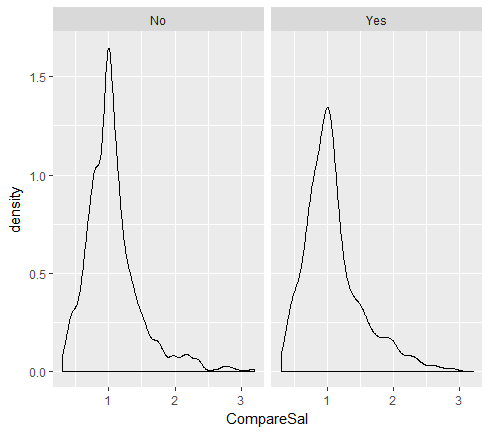
Residual deviance: 2595.1436 on 2938 degrees of freedom

AIC: 2599.1436

Number of Fisher Scoring iterations: 4

# This turns out to be a insignificant variable

ggplot(cart,aes(CompareSal))+geom\_density()+facet\_grid(~Attrition)



**Variable Transformation**

Though cart has the ability to forms group based on classification power, we will create buckets for few variables which are intuitively similar. This avoids too many peaks in cart model

cart$AgeGroup <- with(cart,ifelse(Age>55,8,ifelse(Age>50,7,ifelse(Age>45,6,ifelse(Age>40,5,ifelse(Age>35,4,ifelse(Age>30,3,ifelse(Age>25,2,1)))))))) #Creating Age Groups

cart$DistanceGroup <- with(cart,ifelse(DistanceFromHome>25,6,ifelse(DistanceFromHome>20,5,ifelse(DistanceFromHome>15,4,ifelse(DistanceFromHome>10,3,ifelse(DistanceFromHome>5,2,1)))))) #Creating Distance Groups

cart$YearsWithManagerGroup <- with(cart,ifelse(YearsWithCurrManager>15,5,ifelse(YearsWithCurrManager>10,4,ifelse(YearsWithCurrManager>5,3,ifelse(YearsWithCurrManager>2,2,1))))) #Creating YearsWithManager Groups

cart$stability <- with(cart,ifelse(stability>35,9,ifelse(stability>30,8,ifelse(stability>25,7,ifelse(stability>20,6,ifelse(stability>15,5,ifelse(stability>10,4,ifelse(stability>5,3,ifelse(stability>2,2,1))))))))) #Creating Tenure Per Job groups

cart$WorkYearGroup <- with(cart,ifelse(TotalWorkingYears>35,9,ifelse(TotalWorkingYears>30,8,ifelse(TotalWorkingYears>25,7,ifelse(TotalWorkingYears>20,6,ifelse(TotalWorkingYears>15,5,ifelse(TotalWorkingYears>10,4,ifelse(TotalWorkingYears>5,3,ifelse(TotalWorkingYears>2,2,1)))))))))

cart$NumCompGroup <- with(cart,ifelse(NumCompaniesWorked>4,3,ifelse(NumCompaniesWorked>2,2,1))) #Creating Number of Companies Worked

cart$DailyRateGroup <- with(cart,ifelse(DailyRate>=1250,6,ifelse(DailyRate>1000,5,ifelse(DailyRate>750,4,ifelse(DailyRate>500,3,ifelse(DailyRate > 250,2,1))))))

cart$HourlyRateGroup <- with(cart,ifelse(HourlyRate>=90,7,ifelse(HourlyRate>80,6,ifelse(HourlyRate>70,5,ifelse(HourlyRate>60,4,ifelse(HourlyRate > 50,3,ifelse(HourlyRate>40,2,1)))))))

cart$MonthlyRateGroup <- with(cart,ifelse(MonthlyRate>=25000,6,ifelse(MonthlyRate>20000,5,ifelse(MonthlyRate>15000,4,ifelse(MonthlyRate>10000,3,ifelse(MonthlyRate > 5000,2,1))))))

cart$MonthlyIncomeGroup <- with(cart,ifelse(MonthlyIncome>=18000,7,ifelse(MonthlyIncome>15000,6,ifelse(MonthlyIncome>12000,5,ifelse(MonthlyIncome>9000,4,ifelse(MonthlyIncome > 6000,3,ifelse(MonthlyIncome>3000,2,1)))))))

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

set.seed(1234)

cart$random <- runif(nrow(cart), 0, 1);

cart <- cart[order(cart$random),]

#SEPARATE DATA BASED ON VALUE OF RANDOM COLUMN

cart.dev <- cart[which(cart$random <= 0.7),]

cart.val <- cart[which(cart$random > 0.7),]

#SHOWS ROWCOUNT FOR DEV AND VALIDATION SAMPLE

c(nrow(cart.dev), nrow(cart.val))

[1] 2082 858

length(which(cart.dev$Attrition=="Yes"))/nrow(cart.dev)

[1] 0.1671469741

length(which(cart.val$Attrition=="Yes"))/nrow(cart.val)

[1] 0.1468531469

# remove the random variable

cart.dev = subset(cart.dev, select = -random)

cart.val = subset(cart.val, select = -random)

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

**These would be done under the following heads**

* **Data itself has no explanatory power, that is zero variance**
* **Variables that have been transformed like buckets or any other form**
* **Variables that have shown high p-values or are insignificant**

***Zero Variance variables***

library(caret)

nsv <- nearZeroVar(cart, saveMetrics=TRUE)

nsv <-cbind("ColNo"=1:ncol(cart),nsv)

nsv # this shows standard hours has just one value, so remove this as well

cart = subset(cart, select = c(-EmployeeCount, -Over18,-StandardHours,-EmployeeNumber))

*Remove transformed variables*

cart = subset(cart, select = c(-NumCompaniesWorked,-DistanceFromHome,-Age,-YearsWithCurrManager,-TotalWorkingYears, -DailyRate,-HourlyRate,-MonthlyRate,-MonthlyIncome))

*Remove insignificant variables*

cart = subset(cart, select = c(-Gender,-PerformanceRating,-CompareSal))

**Build CART Model**

library(rpart)

library(rpart.plot)

cartParameters = rpart.control(minsplit=30, minbucket = 10, cp = 0, xval = 10)

cartModel <- rpart(formula = Attrition ~ ., data = cart.dev, method = "class", control = cartParameters)

library(rattle)

library(RColorBrewer)

fancyRpartPlot(cartModel)

printcp(cartModel)

Classification tree:

rpart(formula = Attrition ~ ., data = cart.dev, method = "class",

control = cartParameters)

Variables actually used in tree construction:

[1] BusinessTravel DailyRateGroup DistanceGroup EnvironmentSatisfaction

[5] JobInvolvement JobLevel JobRole OverTime

[9] stability StockOptionLevel WorkLifeBalance WorkYearGroup

[13] YearsAtCompany

Root node error: 348/2082 = 0.16714697

n= 2082

CP nsplit rel error xerror xstd

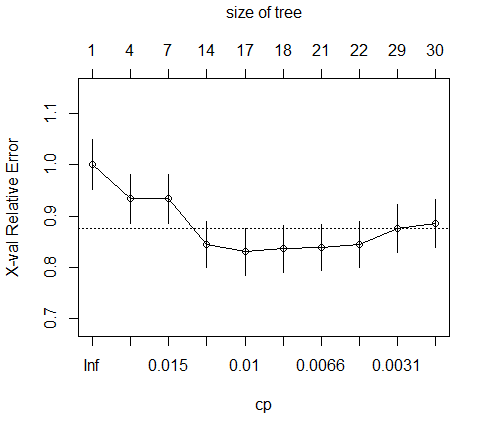
1 0.039272031 0 1.00000000 1.00000000 0.048920914

2 0.017241379 3 0.88218391 0.93390805 0.047589151

3 0.012931034 6 0.83045977 0.93390805 0.047589151

4 0.011494253 13 0.71551724 0.84482759 0.045660201

5 0.008620700 16 0.67528736 0.83045977 0.045333522



Prune the tree to remove overfitting

ptree<- prune(cartModel, cp= 0.0086207,"CP")

printcp(ptree)

Classification tree:

rpart(formula = Attrition ~ ., data = cart.dev, method = "class",

control = cartParameters)

Variables actually used in tree construction:

[1] BusinessTravel DailyRateGroup DistanceGroup EnvironmentSatisfaction

[5] JobInvolvement JobLevel JobRole OverTime

[9] stability StockOptionLevel WorkLifeBalance WorkYearGroup

[13] YearsAtCompany

Root node error: 348/2082 = 0.16714697

n= 2082

CP nsplit rel error xerror xstd

1 0.039272031 0 1.00000000 1.00000000 0.048920914

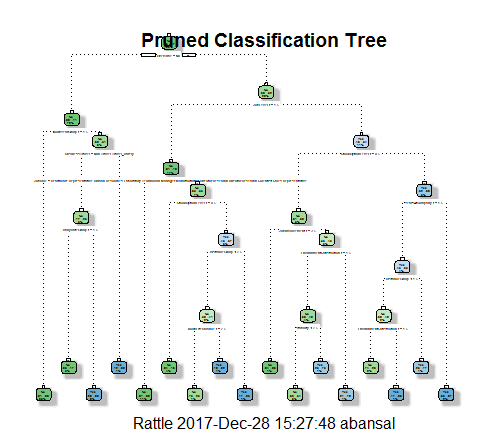
2 0.017241379 3 0.88218391 0.93390805 0.047589151

3 0.012931034 6 0.83045977 0.93390805 0.047589151

4 0.011494253 13 0.71551724 0.84482759 0.045660201

5 0.008620700 16 0.67528736 0.83045977 0.045333522

Final Tree



**Validate CART Model**

**Validation code:**

**Decile and rank ordering**

**## deciling code**

**decile <- function(x){**

**deciles <- vector(length=10)**

**for (i in seq(0.1,1,.1)){**

**deciles[i\*10] <- quantile(x, i, na.rm=T)**

**}**

**return (**

**ifelse(x<deciles[1], 1,**

**ifelse(x<deciles[2], 2,**

**ifelse(x<deciles[3], 3,**

**ifelse(x<deciles[4], 4,**

**ifelse(x<deciles[5], 5,**

**ifelse(x<deciles[6], 6,**

**ifelse(x<deciles[7], 7,**

**ifelse(x<deciles[8], 8,**

**ifelse(x<deciles[9], 9, 10**

**))))))))))**

**};**

**## Ranking code**

**#install.packages("data.table")**

**library(data.table)**

**tmp\_DT = data.table(cart.dev)**

**rank <- tmp\_DT[, list(**

**cnt = length(Attrition),**

**cnt\_resp = length(which(Attrition == 'Yes')),**

**cnt\_non\_resp = length(which(Attrition == 'No'))) ,**

**by=deciles][order(-deciles)];**

**rank$rrate <- round(rank$cnt\_resp \* 100 / rank$cnt,2);**

**rank$cum\_resp <- cumsum(rank$cnt\_resp)**

**rank$cum\_non\_resp <- cumsum(rank$cnt\_non\_resp)**

**rank$cum\_perct\_resp <- round(rank$cum\_resp \* 100 / sum(rank$cnt\_resp),2);**

**rank$cum\_perct\_non\_resp <- round(rank$cum\_non\_resp \* 100 / sum(rank$cnt\_non\_resp),2);**

**rank$ks <- abs(rank$cum\_perct\_resp - rank$cum\_perct\_non\_resp);**

**View(rank)**

**#install.packages("ROCR")**

**library(ROCR)**

**pred <- prediction(cart.dev$predict.score[,2], cart.dev$Attrition)**

**perf <- performance(pred, "tpr", "fpr")**

**plot(perf)**

**KS <- max(attr(perf, 'y.values')[[1]]-attr(perf, 'x.values')[[1]])**

**auc <- performance(pred,"auc");**

**auc <- as.numeric(auc@y.values)**

**#install.packages("ineq")**

**library(ineq)**

**gini = ineq(cart.dev$predict.score[,2], type="Gini")**

**with(cart.dev, table(Attrition, predict.class))**

**auc**

**KS**

**gini**

**Performance on dev data set**

**Performance measure on dev model**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **deciles** | **cnt** | **cnt\_resp** | **cnt\_non\_resp** | **rrate** | **cum\_resp** | **cum\_non\_resp** | **cum\_perct\_resp** | **cum\_perct\_non\_resp** | **ks** |
|  |  |  |  |  |  |  |  |  |  |
| 10 | 236 | 163 | 73 | 69.07 | 163 | 73 | 46.84 | 4.21 | 42.63 |
| 9 | 238 | 45 | 193 | 18.91 | 208 | 266 | 59.77 | 15.34 | 44.43 |
| 8 | 1370 | 120 | 1250 | 8.76 | 328 | 1516 | 94.25 | 87.43 | 6.82 |
| 2 | 221 | 19 | 202 | 8.6 | 347 | 1718 | 99.71 | 99.08 | 0.63 |
| 1 | 17 | 1 | 16 | 5.88 | 348 | 1734 | 100 | 100 | 0 |

with(cart.dev, table(Attrition, predict.class))

predict.class

Attrition No Yes

No 1708 26

Yes 209 139

auc

[1] 0.7532422874

KS

[1] 0.4467247345

gini

[1] 0.4218272108

**Performance measure on validation data set**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **deciles** | **cnt** | **cnt\_resp** | **cnt\_non\_resp** | **rrate** | **cum\_resp** | **cum\_non\_resp** | **cum\_perct\_resp** | **cum\_perct\_non\_resp** | **ks** |
| 10 | 94 | 55 | 39 | 58.51 | 55 | 39 | 43.65 | 5.33 | 38.32 |
| 9 | 86 | 19 | 67 | 22.09 | 74 | 106 | 58.73 | 14.48 | 44.25 |
| 8 | 562 | 42 | 520 | 7.47 | 116 | 626 | 92.06 | 85.52 | 6.54 |
| 2 | 111 | 9 | 102 | 8.11 | 125 | 728 | 99.21 | 99.45 | 0.24 |
| 1 | 5 | 1 | 4 | 20 | 126 | 732 | 100 | 100 | 0 |

**Thus we can observe that now first two deciles have around 59% of attrition, which in intital data set was around 16%. Hence there is considerable improvement in model**

with(cart.val, table(Attrition, predict.class))

predict.class

Attrition No Yes

No 716 16

Yes 83 43

auc

[1] 0.7372007546

KS

[1] 0.4435336976

gini

[1] 0.4018972767

**Build and Validate Neural Network Model**

* **For neural network we need to have data in integer format, hence we have used model.matrix for all categorical variables. Eg:**

**data.frame(NNDF,model.matrix(~BusinessTravel-1))**

* **All ordinal variables like Education, Job level were converted to dummy variables**
* **All Numerical variables were scaled**
* **Same dev and validation data set were used as that in cart, which would help in ensemble modelling**

**NeuralNet with all variables (2 layers with 9,3 nodes)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **deciles** | **cnt** | **cnt\_resp** | **cnt\_non\_resp** | **rrate** | **cum\_resp** | **cum\_non\_resp** | **cum\_perct\_resp** | **cum\_perct\_non\_resp** | **ks** |
|  |  |  |  |  |  |  |  |  |  |  |
| **1** | 10 | 209 | 207 | 2 | 99% | 207 | 2 | 59% | 0% | 0.59 |
| **2** | 9 | 208 | 132 | 76 | 63% | 339 | 78 | 97% | 4% | 0.93 |
| **3** | 8 | 208 | 1 | 207 | 0% | 340 | 285 | 98% | 16% | 0.82 |
| **4** | 7 | 208 | 0 | 208 | 0% | 340 | 493 | 98% | 28% | 0.7 |
| **5** | 6 | 208 | 0 | 208 | 0% | 340 | 701 | 98% | 40% | 0.58 |
| **6** | 5 | 208 | 0 | 208 | 0% | 340 | 909 | 98% | 52% | 0.46 |
| **7** | 4 | 208 | 4 | 204 | 2% | 344 | 1113 | 99% | 64% | 0.35 |
| **8** | 3 | 209 | 1 | 208 | 0% | 345 | 1321 | 99% | 76% | 0.23 |
| **9** | 2 | 207 | 3 | 204 | 1% | 348 | 1525 | 100% | 88% | 0.12 |
| **10** | 1 | 209 | 0 | 209 | 0% | 348 | 1734 | 100% | 100% | 0 |

with(NNDF.dev, table(misClassTable$Target, misClassTable$Classification))

0 1

0 1732 2

1. 9 339

auc

[1] 0.9825216429

KS

[1] 0.9729845285

gini

[1] 0.8333357832

**Performance on test sample**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **deciles** | **cnt** | **cnt\_resp** | **cnt\_non\_resp** | **rrate** | **cum\_resp** | **cum\_non\_resp** | **cum\_perct\_resp** | | **cum\_perct\_non\_resp** | **ks** |
| **1** | 10 | 87 | 69 | 18 | 79% | 69 | 18 | 55% | 2% | | 0.53 |
| **2** | 9 | 85 | 46 | 39 | 54% | 115 | 57 | 91% | 8% | | 0.83 |
| **3** | 8 | 86 | 3 | 83 | 3% | 118 | 140 | 94% | 19% | | 0.75 |
| **4** | 7 | 85 | 0 | 85 | 0% | 118 | 225 | 94% | 31% | | 0.63 |
| **5** | 6 | 87 | 2 | 85 | 2% | 120 | 310 | 95% | 42% | | 0.53 |
| **6** | 5 | 85 | 0 | 85 | 0% | 120 | 395 | 95% | 54% | | 0.41 |
| **7** | 4 | 85 | 2 | 83 | 2% | 122 | 478 | 97% | 65% | | 0.32 |
| **8** | 3 | 86 | 3 | 83 | 3% | 125 | 561 | 99% | 77% | | 0.22 |
| **9** | 2 | 86 | 1 | 85 | 1% | 126 | 646 | 100% | 88% | | 0.12 |
| **10** | 1 | 86 | 0 | 86 | 0% | 126 | 732 | 100% | 100% | | 0 |

**This model shows great improvement over cart, here first two deciles holds for around 91% of attrition**

with(NNDF.val, table(TestmisClassTable$Target, TestmisClassTable$Classification))

0 1

0 702 30

1 11 115

auc

[1] 0.9423085263

KS

[1] 0.8744470466

gini

[1] 0.827711665

**Let us try to rebuild the model, but will use the variables used in cart.**

[1] BusinessTravel DailyRateGroup DistanceGroup EnvironmentSatisfaction

[5] JobInvolvement JobLevel JobRole OverTime

[9] stability StockOptionLevel WorkLifeBalance WorkYearGroup

[13] YearsAtCompany

Performance on dev sample

with(NNDF.dev, table(misClassTable$Target, misClassTable$Classification))

0 1

0 1728 6

1 42 306

auc

[1] 0.9351907091

KS

[1] 0.8758501372

gini

[1] 0.8343270554

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **deciles** | **cnt** | **cnt\_resp** | **cnt\_non\_resp** | **rrate** | **cum\_resp** | **cum\_non\_resp** | **cum\_perct\_resp** | **cum\_perct\_non\_resp** | **ks** |
| **1** | 10 | 209 | 203 | 6 | 97% | 203 | 6 | 58% | 0% | 0.58 |
| **2** | 9 | 209 | 108 | 101 | 52% | 311 | 107 | 89% | 6% | 0.83 |
| **3** | 8 | 207 | 0 | 207 | 0% | 311 | 314 | 89% | 18% | 0.71 |
| **4** | 7 | 208 | 2 | 206 | 1% | 313 | 520 | 90% | 30% | 0.6 |
| **5** | 6 | 208 | 5 | 203 | 2% | 318 | 723 | 91% | 42% | 0.49 |
| **6** | 5 | 209 | 9 | 200 | 4% | 327 | 923 | 94% | 53% | 0.41 |
| **7** | 4 | 207 | 7 | 200 | 3% | 334 | 1123 | 96% | 65% | 0.31 |
| **8** | 3 | 209 | 4 | 205 | 2% | 338 | 1328 | 97% | 77% | 0.2 |
| **9** | 2 | 208 | 8 | 200 | 4% | 346 | 1528 | 99% | 88% | 0.11 |
| **10** | 1 | 208 | 2 | 206 | 1% | 348 | 1734 | 100% | 100% | 0 |

Performance on test sample

with(NNDF.val, table(TestmisClassTable$Target, TestmisClassTable$Classification))

0 1

0 718 14

1. 24 102

auc

[1] 0.9228467343

KS

[1] 0.804124382

gini

[1] 0.8444272648

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **deciles** | **cnt** | **cnt\_resp** | **cnt\_non\_resp** | **rrate** | **cum\_resp** | **cum\_non\_resp** | **cum\_perct\_resp** | **cum\_perct\_non\_resp** | **ks** |
| **1** | 10 | 86 | 76 | 10 | 88% | 76 | 10 | 60% | 1% | 0.59 |
| **2** | 9 | 86 | 33 | 53 | 38% | 109 | 63 | 87% | 9% | 0.78 |
| **3** | 8 | 86 | 2 | 84 | 2% | 111 | 147 | 88% | 20% | 0.68 |
| **4** | 7 | 86 | 2 | 84 | 2% | 113 | 231 | 90% | 32% | 0.58 |
| **5** | 6 | 86 | 3 | 83 | 3% | 116 | 314 | 92% | 43% | 0.49 |
| **6** | 5 | 85 | 3 | 82 | 4% | 119 | 396 | 94% | 54% | 0.4 |
| **7** | 4 | 85 | 1 | 84 | 1% | 120 | 480 | 95% | 66% | 0.29 |
| **8** | 3 | 86 | 2 | 84 | 2% | 122 | 564 | 97% | 77% | 0.2 |
| **9** | 2 | 86 | 2 | 84 | 2% | 124 | 648 | 98% | 89% | 0.09 |
| **10** | 1 | 86 | 2 | 84 | 2% | 126 | 732 | 100% | 100% | 0 |

This model though has lower figures for auc, ks and gini. But overall this looks to be a better fit as this shows more consistence in dev and holdout samples

Also first two deciles accounts for around 87% of attrition

**Combine NN and CART into Ensemble Model**

* **New data frame is created which contains the scores for prediction from both neuralnet (all variables) and cart**
* **Average of the probabilities were calculated**
* **Depending on probability greater then .5, predictions were found**
* **Same error codes and metrics were then applied on resulting data frame for performance measures**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ensembleDataFrame = as.data.frame(cbind(cart.dev$predict.score[,2],NNDF.dev$Predict.score,NNDF.dev$Attrition))  colnames(ensembleDataFrame) = c("cartScore","NnetScore", "Target")  ensembleDataFrame$AvgScore = (ensembleDataFrame$cartScore+ensembleDataFrame$NnetScore)/2  ensemblemisClassTable = data.frame(Target = ensembleDataFrame$Target, TestPrediction = ensembleDataFrame$AvgScore )  ensemblemisClassTable$Classification = ifelse(ensemblemisClassTable$TestPrediction>0.500,1,0)  Performance on dev model   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | **deciles** | **cnt** | **cnt\_resp** | **cnt\_non\_resp** | **rrate** | **cum\_resp** | **cum\_non\_resp** | **cum\_perct\_resp** | **cum\_perct\_non\_resp** | **ks** | | **1** | 10 | 86 | 80 | 6 | 93% | 80 | 6 | 63% | 1% | 0.62 | | **2** | 9 | 86 | 25 | 61 | 29% | 105 | 67 | 83% | 9% | 0.74 | | **3** | 8 | 87 | 6 | 81 | 7% | 111 | 148 | 88% | 20% | 0.68 | | **4** | 7 | 84 | 2 | 82 | 2% | 113 | 230 | 90% | 31% | 0.59 | | **5** | 6 | 86 | 3 | 83 | 3% | 116 | 313 | 92% | 43% | 0.49 | | **6** | 5 | 86 | 3 | 83 | 3% | 119 | 396 | 94% | 54% | 0.4 | | **7** | 4 | 86 | 2 | 84 | 2% | 121 | 480 | 96% | 66% | 0.3 | | **8** | 3 | 85 | 1 | 84 | 1% | 122 | 564 | 97% | 77% | 0.2 | | **9** | 2 | 87 | 3 | 84 | 3% | 125 | 648 | 99% | 89% | 0.1 | | **10** | 1 | 85 | 1 | 84 | 1% | 126 | 732 | 100% | 100% | 0 |   with(ensembleDataFrame, table(ensemblemisClassTable$Target, ensemblemisClassTable$Classification))    0 1  0 1728 6  1 47 301  auc  [1] 0.9398474062  KS  [1] 0.8741200334  gini  [1] 0.6077289342 |
|  |
|  |
| |  | | --- | | Performace on validation data set |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | **deciles** | **cnt** | **cnt\_resp** | **cnt\_non\_resp** | **rrate** | **cum\_resp** | **cum\_non\_resp** | **cum\_perct\_resp** | **cum\_perct\_non\_resp** | **ks** | | **1** | 10 | 86 | 80 | 6 | 93% | 80 | 6 | 63% | 1% | 0.62 | | **2** | 9 | 86 | 25 | 61 | 29% | 105 | 67 | 83% | 9% | 0.74 | | **3** | 8 | 87 | 6 | 81 | 7% | 111 | 148 | 88% | 20% | 0.68 | | **4** | 7 | 84 | 2 | 82 | 2% | 113 | 230 | 90% | 31% | 0.59 | | **5** | 6 | 86 | 3 | 83 | 3% | 116 | 313 | 92% | 43% | 0.49 | | **6** | 5 | 86 | 3 | 83 | 3% | 119 | 396 | 94% | 54% | 0.4 | | **7** | 4 | 86 | 2 | 84 | 2% | 121 | 480 | 96% | 66% | 0.3 | | **8** | 3 | 85 | 1 | 84 | 1% | 122 | 564 | 97% | 77% | 0.2 | | **9** | 2 | 87 | 3 | 84 | 3% | 125 | 648 | 99% | 89% | 0.1 | | **10** | 1 | 85 | 1 | 84 | 1% | 126 | 732 | 100% | 100% | 0 |   with(ensembleDataFrame, table(ensemblemisClassTable$Target, ensemblemisClassTable$Classification))    0 1  0 718 14  1 27 99  auc  [1] 0.9219143031  KS  [1] 0.7868852459  gini  [1] 0.5926827237  These results are more inclined towards the output of neural net. |

**Model comparison for cart, neuralnet and ensemble**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Cart | | NN(all variable) | | NN(Selected variables) | | Ensemle | |
|  | Dev | Test | Dev | Test | Dev | Test | Dev | Test |
| AUC | 0.753242287 | 0.737200755 | 0.982521643 | 0.942308526 | 0.935190709 | 0.922846734 | 0.939847406 | 0.921914303 |
| KS | 0.446724735 | 0.443533698 | 0.972984529 | 0.874447047 | 0.875850137 | 0.804124382 | 0.874120033 | 0.786885246 |
| Gini | 0.421827211 | 0.401897277 | 0.833335783 | 0.827711665 | 0.834327055 | 0.844427265 | 0.607728934 | 0.592682724 |

On comparison of different models, we could see that cart gives the lowest measures for performance matrix. Neural net with all variables outperforms on this aspect.

But if we look at neural net model with selected variables (Variables used in cart model) we do get the better model in terms of consistency in dev and holdout samples.

Simple averaging out the probability and prediction is better than cart but again we would choose the third model.

Let us also try stacking algorithms from caretEnsemble package. We would also include neural net with model averaging to check for any improvement

library(caretEnsemble)

control <- trainControl(method="repeatedcv", number=10, repeats=3, savePredictions=TRUE, classProbs=TRUE)

algorithmList <- c('rpart', 'nnet', 'avNNet')

# #set.seed(seed)

models <- caretList(Attrition~., data=cart, trControl=control, methodList=algorithmList)

results <- resamples(models)

summary(results)

Call:

summary.resamples(object = results)

Models: rpart, nnet, avNNet

Number of resamples: 30

Accuracy

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

rpart 0.8265306122 0.8435374150 0.8505937968 0.8494358730 0.8562925170 0.8673469388 0

nnet 0.8639455782 0.8874677857 0.8979591837 0.8997762347 0.9084745763 0.9489795918 0

avNNet 0.8741496599 0.9045181212 0.9131201969 0.9114562317 0.9183673469 0.9489795918 0

Kappa

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

rpart 0.1187969925 0.2417280990 0.2625261825 0.2736618395 0.3079664254 0.4214350590 0

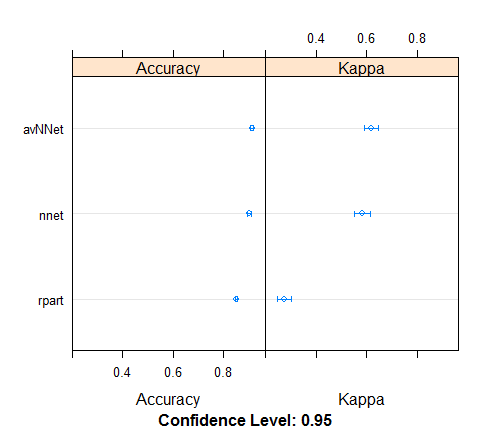
nnet 0.4017094017 0.5270844409 0.5773123725 0.5825797652 0.6251650896 0.7978733156 0

avNNet 0.4628148148 0.5734245238 0.6200182302 0.6175074791 0.6542432884 0.7904191617 0

0

We could see that accuracy is now 84.7 from cart and 89.98 from nnet

dotplot(results)



We could see that nnet and avnnet are similar and outperforms cart both in accuracy and kappa

We could further tune the model by using the algorithms for feature selection and feeding those variables in selected variable neural net to improve on this model.

Also caretEnsemblealso provides ensembling technique by bagging and boosting methods, which could be compared to tune the final model.