**DataMining Assignment**

**on**

# **Attrition-HR Data**

**Problem Statement:**

A Company’s HR is strongminded to reduce the Employee Attrition. They want achieve this by analyzing the employees data, which has lot of professional and some personal information.

**Our Solution:**

We can use the statistical models on the Employee data, which can classify the employee into two classes,

* employee keen on leaving the company (Candidate for Attrition)
* employee happy at work and not keen on leaving the company (Not a Candidate for Attrition)

**How to achieve the solution:**

Analyzing the nature of the problem and data, It is /has

* Classification problem of dividing the employee into two groups
* Large number of independent variables
* Dependent and independent variables are not linearly related
* We are not aware which variables contribute the most

Hence we can try to build quite a few classification models like CART,NN and ENSEMBLE and choose the best model which suits our business case.

**Loading the data:**

##Import Data File

setwd("C:/Users/ksivapra/Desktop/Rtest")

mydata=read.csv("HR\_Employee\_Attrition\_Data.csv",header=TRUE)

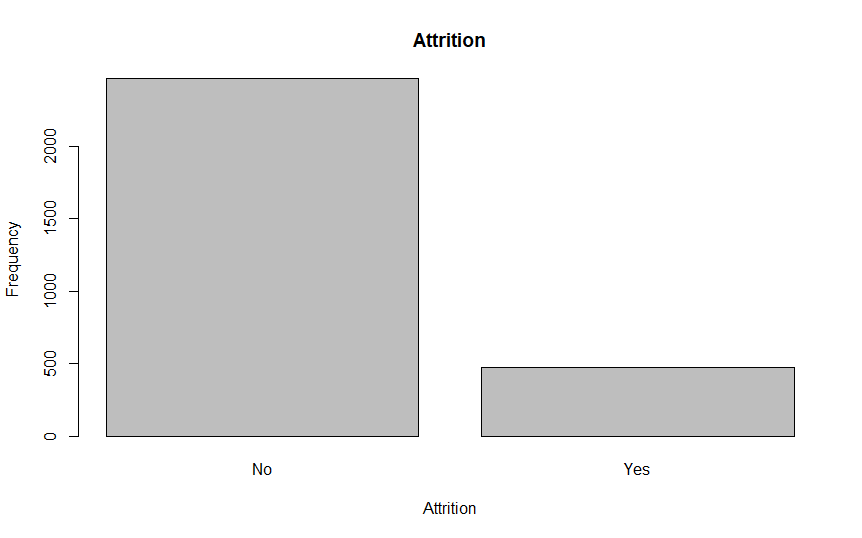
attach(mydata)

mydata

**Data Exploration:**

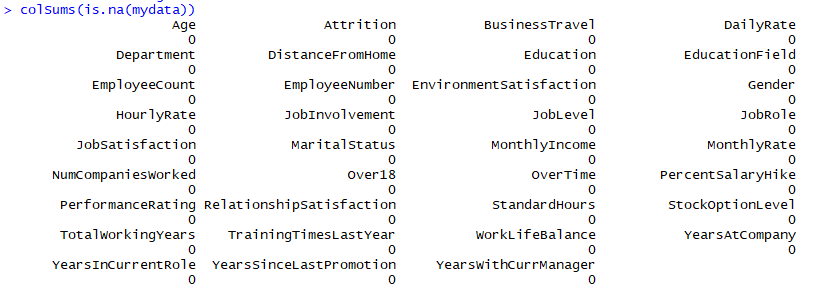
Before we go to modeling of the problem, we would like to explore the data to uncover hidden trends and insight, validate Missing data, look out for outliers and insignificant independent variables.

1. Univariate analysis: Analyze the target variable(Attrition) to highlight missing and outlier values

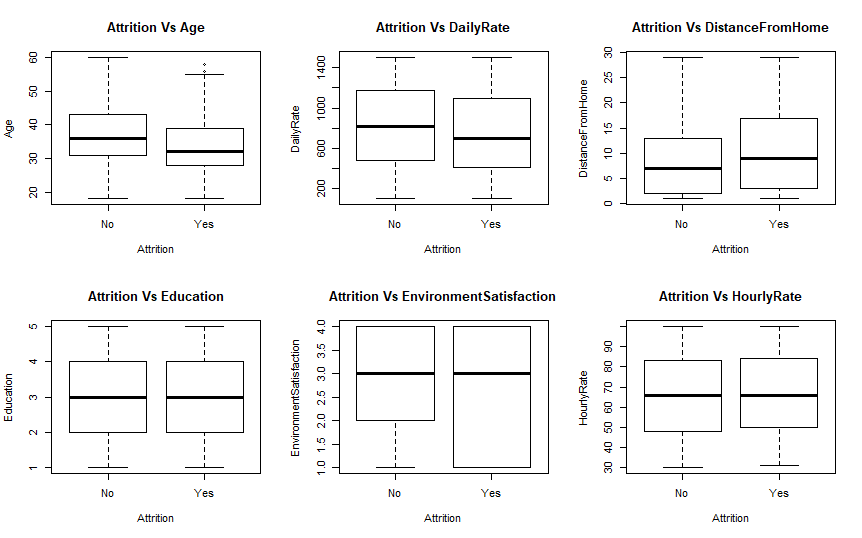


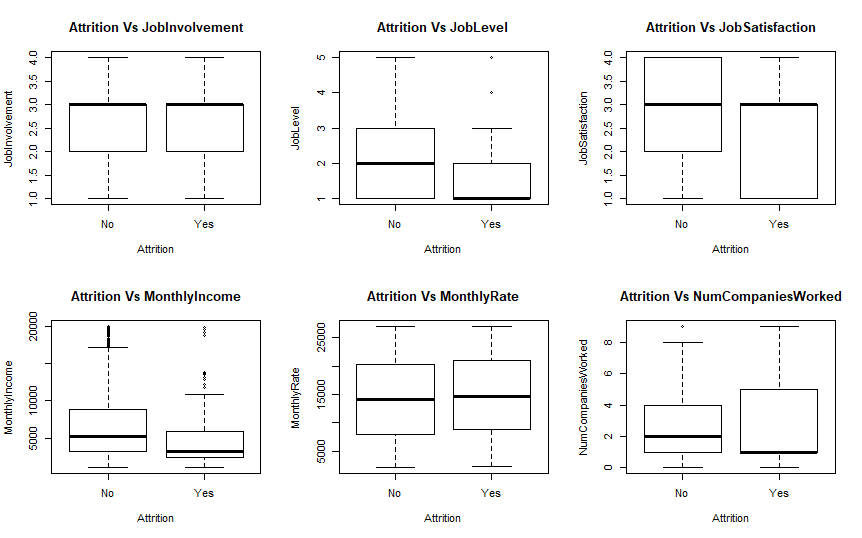
There are no missing values on our target variable.

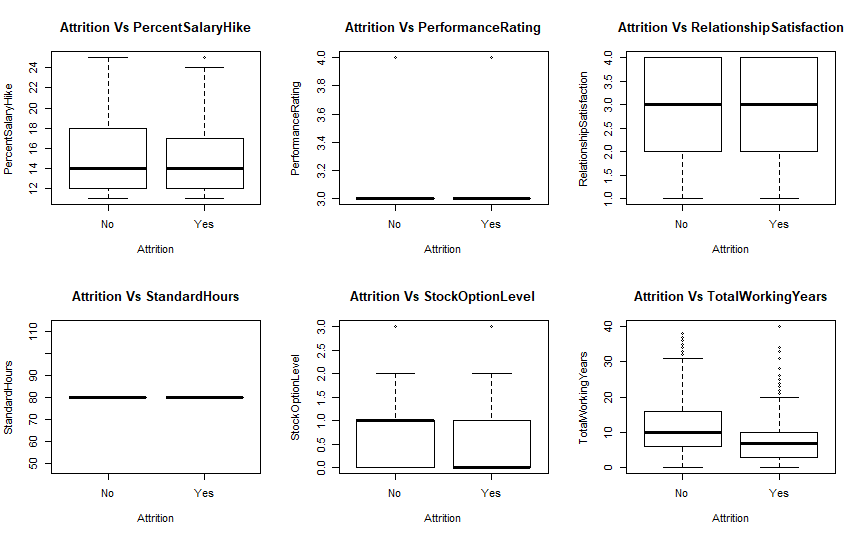
Validate if there is for missing value:

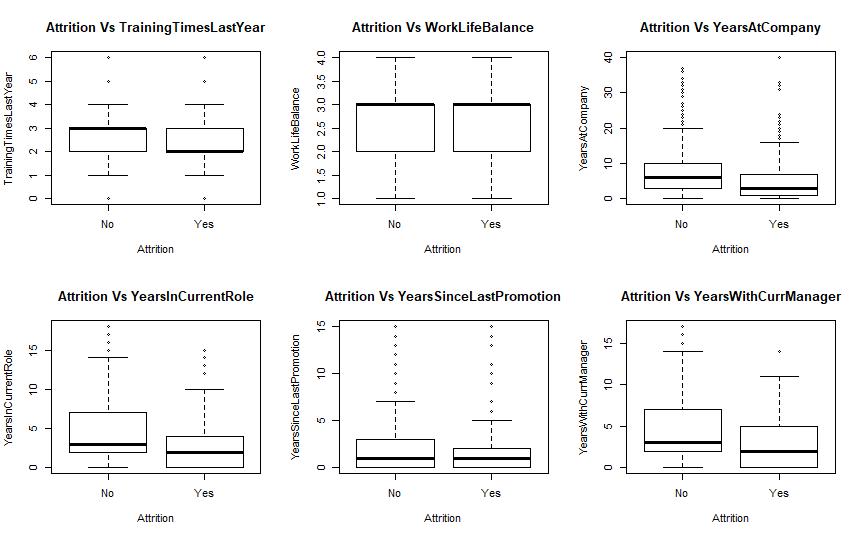


1. Bi-variate Analysis:

We have use a combination of Visual plot & chi-square test find the correlation between our target & independent variable.







**Hypothesis testing:**

**Chi-Square test** to find the correlation between Categorial and categorial variable

**T-test** to find the correlation between Categorial and continuous variable

**Chi-SquaredTesting:**

**Null Hypothesis:**

There is no relationship between the categorical variables that we are testing

**Alternate Hypothesis:**

There is a relationship between categorical variables we are testing

|  |
| --- |
| > # Find corelation: use CHi-square test to understand the corelation of the categorical variable with the target variable  > chisq.test(mydata$Attrition, mydata$BusinessTravel)  Pearson's Chi-squared test  data: mydata$Attrition and mydata$BusinessTravel  X-squared = 41.05204, df = 2, p-value = 0.000000001218043  > chisq.test(mydata$Attrition, mydata$Department)  Pearson's Chi-squared test  data: mydata$Attrition and mydata$Department  X-squared = 17.355912, df = 2, p-value = 0.0001702988  > chisq.test(mydata$Attrition, mydata$EducationField)  Pearson's Chi-squared test  data: mydata$Attrition and mydata$EducationField  X-squared = 32.573181, df = 5, p-value = 0.000004573897  > chisq.test(mydata$Attrition, mydata$Gender)  Pearson's Chi-squared test with Yates' continuity correction  data: mydata$Attrition and mydata$Gender  X-squared = 0.8087734, df = 1, p-value = 0.3684831  > chisq.test(mydata$Attrition, mydata$JobRole)  Pearson's Chi-squared test  data: mydata$Attrition and mydata$JobRole  X-squared = 157.88626, df = 8, p-value < 0.00000000000000022204  > chisq.test(mydata$Attrition, mydata$MaritalStatus)  Pearson's Chi-squared test  data: mydata$Attrition and mydata$MaritalStatus  X-squared = 55.559131, df = 2, p-value = 0.0000000000008619617  > chisq.test(mydata$Attrition, mydata$Over18)  Error in chisq.test(mydata$Attrition, mydata$Over18) :  'x' and 'y' must have at least 2 levels  > chisq.test(mydata$Attrition, mydata$OverTime)  Pearson's Chi-squared test with Yates' continuity correction  data: mydata$Attrition and mydata$OverTime  X-squared = 146.09715, df = 1, p-value < 0.00000000000000022204  > chisq.test(mydata$Attrition, mydata$Education)  Pearson's Chi-squared test  data: mydata$Attrition and mydata$Education  X-squared = 10.001993, df = 4, p-value = 0.04039413  > chisq.test(mydata$Attrition, mydata$EnvironmentSatisfaction)  Pearson's Chi-squared test  data: mydata$Attrition and mydata$EnvironmentSatisfaction  X-squared = 41.768733, df = 3, p-value = 0.000000004492088  > chisq.test(mydata$Attrition, mydata$JobInvolvement)  Pearson's Chi-squared test  data: mydata$Attrition and mydata$JobInvolvement  X-squared = 49.473903, df = 3, p-value = 0.000000000103403  > chisq.test(mydata$Attrition, mydata$JobLevel)  Pearson's Chi-squared test  data: mydata$Attrition and mydata$JobLevel  X-squared = 122.2817, df = 4, p-value < 0.00000000000000022204  > chisq.test(mydata$Attrition, mydata$JobSatisfaction)  Pearson's Chi-squared test  data: mydata$Attrition and mydata$JobSatisfaction  X-squared = 19.254567, df = 3, p-value = 0.0002421838  > chisq.test(mydata$Attrition, mydata$NumCompaniesWorked)  Pearson's Chi-squared test  data: mydata$Attrition and mydata$NumCompaniesWorked  X-squared = 37.359385, df = 9, p-value = 0.00002270201  > chisq.test(mydata$Attrition, mydata$PercentSalaryHike)  Pearson's Chi-squared test  data: mydata$Attrition and mydata$PercentSalaryHike  X-squared = 31.212285, df = 14, p-value = 0.0051763  > chisq.test(mydata$Attrition, mydata$RelationshipSatisfaction)  Pearson's Chi-squared test  data: mydata$Attrition and mydata$RelationshipSatisfaction  X-squared = 9.3441423, df = 3, p-value = 0.02504856  > chisq.test(mydata$Attrition, mydata$StockOptionLevel)  Pearson's Chi-squared test  data: mydata$Attrition and mydata$StockOptionLevel  X-squared = 76.678953, df = 3, p-value < 0.00000000000000022204  > chisq.test(mydata$Attrition, mydata$TrainingTimesLastYear)  Pearson's Chi-squared test  data: mydata$Attrition and mydata$TrainingTimesLastYear  X-squared = 21.438241, df = 6, p-value = 0.001529875  > chisq.test(mydata$Attrition, mydata$WorkLifeBalance)  Pearson's Chi-squared test  data: mydata$Attrition and mydata$WorkLifeBalance  X-squared = 26.728634, df = 3, p-value = 0.000006711206  **T Test:**  **Null Hypothesis:**  There is no significant difference between the sample means of the variables we are testing  **Alternate Hypothesis:**  There is a significant difference between the sample means of the variables we are testing  > t.test(mydata$Age~ mydata$Attrition)  Welch Two Sample t-test  data: mydata$Age by mydata$Attrition  t = 7.3088817, df = 502.58784, p-value = 0.000000000001066857  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  2.858582798 4.960397276  sample estimates:  mean in group No mean in group Yes  37.56028369 33.65079365  > t.test(mydata$DailyRate~ mydata$Attrition)  Welch Two Sample t-test  data: mydata$DailyRate by mydata$Attrition  t = 2.6080127, df = 534.02493, p-value = 0.009361852  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  14.51731558 103.13820620  sample estimates:  mean in group No mean in group Yes  814.1610942 755.3333333  > t.test(mydata$DistanceFromHome~ mydata$Attrition)  Welch Two Sample t-test  data: mydata$DistanceFromHome by mydata$Attrition  t = -4.0772838, df = 509.03921, p-value = 0.00005286392  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  -2.875784667 -1.005561725  sample estimates:  mean in group No mean in group Yes  8.879432624 10.820105820  > t.test(mydata$HourlyRate~ mydata$Attrition)  Welch Two Sample t-test  data: mydata$HourlyRate by mydata$Attrition  t = 1.9042945, df = 544.0458, p-value = 0.05739904  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  -0.06662035451 4.29267011247  sample estimates:  mean in group No mean in group Yes  66.26646403 64.15343915  > t.test(mydata$MonthlyIncome~ mydata$Attrition)  Welch Two Sample t-test  data: mydata$MonthlyIncome by mydata$Attrition  t = 10.452892, df = 684.01666, p-value < 0.00000000000000022204  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  1788.932058 2616.419287  sample estimates:  mean in group No mean in group Yes  6811.511651 4608.835979  > t.test(mydata$MonthlyRate~ mydata$Attrition)  Welch Two Sample t-test  data: mydata$MonthlyRate by mydata$Attrition  t = -0.23732978, df = 525.48708, p-value = 0.8124935  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  -885.4217996 694.5459693  sample estimates:  mean in group No mean in group Yes  14199.51976 14294.95767  > t.test(mydata$PerformanceRating~ mydata$Attrition)  Welch Two Sample t-test  data: mydata$PerformanceRating by mydata$Attrition  t = 0.77290689, df = 544.58556, p-value = 0.4399128  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  -0.02429445032 0.05581533291  sample estimates:  mean in group No mean in group Yes  3.169199595 3.153439153  > t.test(mydata$TotalWorkingYears~ mydata$Attrition)  Welch Two Sample t-test  data: mydata$TotalWorkingYears by mydata$Attrition  t = 8.8865943, df = 543.53426, p-value < 0.00000000000000022204  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  2.876687567 4.509330670  sample estimates:  mean in group No mean in group Yes  11.88348531 8.19047619  > t.test(mydata$YearsAtCompany~ mydata$Attrition)  Welch Two Sample t-test  data: mydata$YearsAtCompany by mydata$Attrition  t = 6.6265722, df = 530.92799, p-value = 0.00000000008451604  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  1.592909289 2.935290643  sample estimates:  mean in group No mean in group Yes  7.391084093 5.126984127  > t.test(mydata$YearsInCurrentRole~ mydata$Attrition)  Welch Two Sample t-test  data: mydata$YearsInCurrentRole by mydata$Attrition  t = 8.750856, df = 591.55756, p-value < 0.00000000000000022204  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  1.233153893 1.946852862  sample estimates:  mean in group No mean in group Yes  4.510638298 2.920634921  > t.test(mydata$YearsSinceLastPromotion~ mydata$Attrition)  Welch Two Sample t-test  data: mydata$YearsSinceLastPromotion by mydata$Attrition  t = 1.3632634, df = 539.66034, p-value = 0.173368  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  -0.1079123753 0.5973866520  sample estimates:  mean in group No mean in group Yes  2.255319149 2.010582011  > t.test(mydata$YearsWithCurrManager~ mydata$Attrition)  Welch Two Sample t-test  data: mydata$YearsWithCurrManager by mydata$Attrition  t = 9.4628411, df = 596.99774, p-value < 0.00000000000000022204  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  1.343153657 2.046692116  sample estimates:  mean in group No mean in group Yes  4.435663627 2.740740741 |
| **DROPPING the non-Significant columns from Modeling:** |
|  |

Based on our Analysis (statistical and visual methods), the below variable are not significance to the outcome of the target variable. So, we drop them from our analysis.

1. Over18.
2. EmployeeCount.
3. EmployeeNumber.
4. PerformanceRating.
5. StandardHours

**Dummy coding for categorical Variables:**

Here we are converting the categorical variables into numerical by dummy hot coding

#One hot coding

codedd<-onehot(mydata)

View(codedd)

dnew<-predict(codedd,mydata)

as.data.frame(dnew) ->dnew

**Tranforming the data to Z value:**

Here we are converting all the variables with different values into Z score for uniformity.

Here I have don’t it for few , we need to do the same for all required variables

dnew$Age <- round(dnew$Age/10)

dnew$DailyRate <- round((dnew$DailyRate-mean(dnew$DailyRate))/sd(dnew$DailyRate))

dnew$HourlyRate <- round((dnew$DailyRate-mean(dnew$DailyRate))/sd(dnew$DailyRate))

dnew$MonthlyIncome<- round((dnew$MonthlyIncome-mean(dnew$MonthlyIncome))/sd(dnew$MonthlyIncome))

**Dividing data into Trainingdata and Testdata:**

Dividing the Trainingdata and Test data in 70:30 ratio

smp\_size <- floor(0.70 \* nrow(dnewfinal))

id <-sample.int(n=nrow(dnewfinal),size = smp\_size,replace = F)

traindata <- dnewfinal[id,]

View(traindata)

testdata <- dnewfinal[-id,]

View(testdata)

**MODEL BUIDING:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DECISION TREE:**  Model building with Train data:  > rpart(Attrition ~ .,data=traindata,control=rpart.control(cp=0.001),method="class") -> dtree  > rpart.plot((dtree)   |  | | --- | | > printcp(dtree)  Classification tree:  rpart(formula = Attrition ~ ., data = traindata, method = "class",  control = rpart.control(cp = 0.001))  Variables actually used in tree construction:  [1] Age BusinessTravel\_Travel\_Frequently  [3] DistanceFromHome EducationField\_Technical\_Degree  [5] EnvironmentSatisfaction Gender\_Female  [7] JobInvolvement JobLevel  [9] JobRole\_Laboratory\_Technician JobRole\_Sales\_Executive  [11] JobSatisfaction MaritalStatus\_Single  [13] NumCompaniesWorked OverTime\_No  [15] PercentSalaryHike RelationshipSatisfaction  [17] StockOptionLevel TotalWorkingYears  [19] TrainingTimesLastYear WorkLifeBalance  [21] YearsAtCompany YearsInCurrentRole  [23] YearsSinceLastPromotion YearsWithCurrManager  Root node error: 330/2058 = 0.16034985  n= 2058  CP nsplit rel error xerror xstd  1 0.0333333333 0 1.00000000 1.00000000 0.050441990  2 0.0212121212 3 0.90000000 0.98484848 0.050130766  3 0.0181818182 7 0.81515152 0.91818182 0.048710574  4 0.0136363636 8 0.79696970 0.88787879 0.048036299  5 0.0121212121 10 0.76969697 0.87878788 0.047830344  6 0.0106060606 12 0.74545455 0.88181818 0.047899187  7 0.0098484848 14 0.72424242 0.86363636 0.047483221  8 0.0090909091 18 0.68484848 0.86363636 0.047483221  9 0.0045454545 25 0.61818182 0.87272727 0.047692079  10 0.0043771044 27 0.60909091 0.89696970 0.048240542  11 0.0037878788 36 0.56969697 0.90303030 0.048375764  12 0.0030303030 40 0.55454545 0.90303030 0.048375764  13 0.0010101010 46 0.53636364 0.91212121 0.048577203  14 0.0010000000 49 0.53333333 0.92121212 0.048776986 | |  | | |  | | --- | | > | | |
| Root node error: 330/2058 = 0.16034985  -Without any model we classified 330 items as 1 out of 2058, which is the misclassification error rate  4 0.0136363636 8 0.79696970 0.88787879 0.048036299  5 0.0121212121 10 0.76969697 0.87878788 0.047830344  6 0.0106060606 12 0.74545455 0.88181818 0.047899187  7 0.0098484848 14 0.72424242 0.86363636 0.047483221  Here we are selecting .013 as a control parameter as line 5 is contained within the X-error of line 4   |  | | --- | | > finaldtree  n= 2058  node), split, n, loss, yval, (yprob)  \* denotes terminal node  1) root 2058 330 1 (0.83965014577 0.16034985423)  2) OverTime\_No>=0.5 1466 147 1 (0.89972714870 0.10027285130) \*  3) OverTime\_No< 0.5 592 183 1 (0.69087837838 0.30912162162)  6) JobLevel>=1.5 370 68 1 (0.81621621622 0.18378378378)  12) JobRole\_Sales\_Executive< 0.5 236 21 1 (0.91101694915 0.08898305085) \*  13) JobRole\_Sales\_Executive>=0.5 134 47 1 (0.64925373134 0.35074626866)  26) MaritalStatus\_Single< 0.5 88 17 1 (0.80681818182 0.19318181818) \*  27) MaritalStatus\_Single>=0.5 46 16 2 (0.34782608696 0.65217391304) \*  7) JobLevel< 1.5 222 107 2 (0.48198198198 0.51801801802)  14) StockOptionLevel>=0.5 113 44 1 (0.61061946903 0.38938053097)  28) BusinessTravel\_Travel\_Frequently< 0.5 83 26 1 (0.68674698795 0.31325301205) \*  29) BusinessTravel\_Travel\_Frequently>=0.5 30 12 2 (0.40000000000 0.60000000000)  58) Gender\_Female< 0.5 22 10 1 (0.54545454545 0.45454545455)  116) NumCompaniesWorked< 2 11 1 1 (0.90909090909 0.09090909091) \*  117) NumCompaniesWorked>=2 11 2 2 (0.18181818182 0.81818181818) \*  59) Gender\_Female>=0.5 8 0 2 (0.00000000000 1.00000000000) \*  15) StockOptionLevel< 0.5 109 38 2 (0.34862385321 0.65137614679)  30) YearsSinceLastPromotion>=-0.5 62 30 1 (0.51612903226 0.48387096774)  60) Age>=3.5 18 2 1 (0.88888888889 0.11111111111) \*  61) Age< 3.5 44 16 2 (0.36363636364 0.63636363636) \*  31) YearsSinceLastPromotion< -0.5 47 6 2 (0.12765957447 0.87234042553) \* | |  | | |  | | --- | | > | | |
|  |
| Cutoff : 0 .1 can be taken as cutoff from this ROCR curve to get a good accuracy |
| Accuracy for Decision tree:  > table(actual=testdata$Attrition,predicted=pred1)  predicted  actual 1 2  1 724 14  2 104 40  > (724+40)/882  [1] 0.8662132  **Accuaracy : 86%** |
| **Regression Tree** |
| Model building with Train data:   |  | | --- | | > rpart(Attrition ~ .,data=traindata,method="anova",control=rpart.control(cp=0.002)) -> regtree  > rpart.plot(regtree)  > printcp(regtree)  Regression tree:  rpart(formula = Attrition ~ ., data = traindata, method = "anova",  control = rpart.control(cp = 0.002))  Variables actually used in tree construction:  [1] Age BusinessTravel=Travel\_Frequently  [3] DailyRate Department=Research & Development  [5] DistanceFromHome Education  [7] EducationField=Technical Degree EnvironmentSatisfaction  [9] Gender=Female JobInvolvement  [11] JobLevel JobRole=Laboratory Technician  [13] JobRole=Manager JobRole=Sales Executive  [15] JobSatisfaction MaritalStatus=Divorced  [17] MaritalStatus=Single MonthlyRate  [19] NumCompaniesWorked OverTime=Yes  [21] PercentSalaryHike RelationshipSatisfaction  [23] StockOptionLevel TotalWorkingYears  [25] TrainingTimesLastYear WorkLifeBalance  [27] YearsAtCompany YearsInCurrentRole  [29] YearsSinceLastPromotion YearsWithCurrManager  Root node error: 277.08/2058 = 0.13464  n= 2058  CP nsplit rel error xerror xstd  1 0.0663838 0 1.00000 1.00073 0.040838  2 0.0559400 1 0.93362 0.94949 0.038752  3 0.0220522 2 0.87768 0.88113 0.038301  4 0.0148926 4 0.83357 0.85221 0.038249  5 0.0141523 7 0.78889 0.84817 0.038407  6 0.0132649 9 0.76059 0.84254 0.038941  7 0.0127191 10 0.74732 0.84886 0.039568  8 0.0108095 11 0.73461 0.84161 0.039655  9 0.0090190 12 0.72380 0.85004 0.040366  10 0.0078709 13 0.71478 0.83635 0.040533  11 0.0078457 17 0.68329 0.84155 0.041145  12 0.0076879 18 0.67545 0.84192 0.041192  13 0.0069933 19 0.66776 0.84660 0.041404  14 0.0063267 20 0.66077 0.86012 0.042123  15 0.0060958 22 0.64811 0.85837 0.042150  16 0.0059802 23 0.64202 0.85983 0.042217  17 0.0059119 30 0.59879 0.85627 0.042248  18 0.0058653 32 0.58696 0.85351 0.042161  19 0.0058127 33 0.58110 0.85351 0.042161  20 0.0058017 36 0.56366 0.85287 0.042166  21 0.0057931 39 0.54625 0.85287 0.042166  22 0.0051557 41 0.53467 0.85390 0.042489  23 0.0045846 42 0.52951 0.84979 0.042421  24 0.0043988 44 0.52034 0.86130 0.042642  25 0.0041808 45 0.51594 0.86578 0.042630  26 0.0036692 49 0.49591 0.85648 0.042290  27 0.0033684 50 0.49224 0.85273 0.042284  28 0.0033219 51 0.48888 0.85142 0.042134  29 0.0032729 52 0.48555 0.85042 0.042119  30 0.0031810 53 0.48228 0.85007 0.042053  31 0.0027841 54 0.47910 0.84756 0.041823  32 0.0025459 56 0.47353 0.85896 0.042018  33 0.0024862 58 0.46844 0.86282 0.042032  34 0.0022328 60 0.46347 0.86647 0.042230  35 0.0022048 61 0.46123 0.86565 0.042193  36 0.0021968 63 0.45682 0.86565 0.042193  37 0.0021132 64 0.45463 0.86609 0.042194  38 0.0020000 65 0.45251 0.86550 0.042041 | |  | | |  | | --- | | > | | |
| Reg Tree Plot: |
| |  | | --- | | > table(actual=testdata$Attrition,predicted=predregtree)  predicted  actual 1 2  1 702 36  2 95 49 | |  | | |  | | --- | | > (702+49)/882  [1] 0.8514739  Accuracy: 85% | | |
|  |

|  |
| --- |
| **Neural network** |
| **Get a Columnstring:**   |  | | --- | | > ## Train NEURAL NETWORK  > library(stringr)  > str\_replace\_all(colnames(traindata),"=","\_") -> colnames(traindata)  > str\_replace\_all(colnames(traindata),"-","\_") -> colnames(traindata)  > str\_replace\_all(colnames(traindata)," ","\_") -> colnames(traindata)  > str\_replace\_all(colnames(traindata),"\_&\_","\_And\_") -> colnames(traindata)  > #For testdata  > str\_replace\_all(colnames(testdata),"=","\_") -> colnames(testdata)  > str\_replace\_all(colnames(testdata),"-","\_") -> colnames(testdata)  > str\_replace\_all(colnames(testdata)," ","\_") -> colnames(testdata)  > str\_replace\_all(colnames(testdata),"\_&\_","\_And\_") -> colnames(testdata)  > ###  > paste(colnames(traindata)[2:52],collapse="+") -> s  > paste("Attrition ~",s) -> strcolnames  > strcolnames  [1] "Attrition ~ Age+BusinessTravel\_Non\_Travel+BusinessTravel\_Travel\_Frequently+BusinessTravel\_Travel\_Rarely+DailyRate+Department\_Human\_Resources+Department\_Research\_And\_Development+Department\_Sales+DistanceFromHome+Education+EducationField\_Human\_Resources+EducationField\_Life\_Sciences+EducationField\_Marketing+EducationField\_Medical+EducationField\_Other+EducationField\_Technical\_Degree+EnvironmentSatisfaction+Gender\_Female+Gender\_Male+HourlyRate+JobInvolvement+JobLevel+JobRole\_Healthcare\_Representative+JobRole\_Human\_Resources+JobRole\_Laboratory\_Technician+JobRole\_Manager+JobRole\_Manufacturing\_Director+JobRole\_Research\_Director+JobRole\_Research\_Scientist+JobRole\_Sales\_Executive+JobRole\_Sales\_Representative+JobSatisfaction+MaritalStatus\_Divorced+MaritalStatus\_Married+MaritalStatus\_Single+MonthlyIncome+MonthlyRate+NumCompaniesWorked+OverTime\_No+OverTime\_Yes+PercentSalaryHike+PerformanceRating+RelationshipSatisfaction+StockOptionLevel+TotalWorkingYears+TrainingTimesLastYear+WorkLifeBalance+YearsAtCompany+YearsInCurrentRole+YearsSinceLastPromotion+YearsWithCurrManager" | |  | | |  | | --- | | > | | |
| **Model building with Train data:**  > nn <- neuralnet(formulastring, data=traindata, hidden=c(5,2), linear.output=TRUE, threshold=0.01) |
| > plot(nn) |
|  |
| nnresults <- compute(nn, testdata[2:52])  > table(actual=testdata$Attrition,predicted=nnresultsroundoff)  predicted  actual 1 2  1 709 29  2 63 81  >> (709+81)/882  [1] 0.89569161  **Accuracy : 89%** |

|  |
| --- |
| **Random Forest** |
| **Model building with Train data:**  > randomForest(Attrition ~ .,data=traindata, ntree=200,mtry=5) -> rforest |
|  |
| |  | | --- | | > predict(rforest,testdata) -> predforest  > table(testdata$Attrition,round(predforest))    1 2  1 738 0  2 43 101 | |  | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | >   |  | | --- | | > (738+101)/882  [1] 0.9512471655 | |  | | |  | | --- | | > **Accuracy : 95%** | | | | |
| > importance(rforest) -> ImpVar  > ImpVar  IncNodePurity  Age 9.8248080696  BusinessTravel\_Non\_Travel 1.0702927893  BusinessTravel\_Travel\_Frequently 4.7202828037  BusinessTravel\_Travel\_Rarely 2.9792206220  DailyRate 6.4036194865  Department\_Human\_Resources 0.6983895412  Department\_Research\_And\_Development 2.4810816488  Department\_Sales 2.8104012757  DistanceFromHome 8.4974622087  Education 7.4113123712  EducationField\_Human\_Resources 0.4325007387  EducationField\_Life\_Sciences 2.6088255527  EducationField\_Marketing 1.7430151007  EducationField\_Medical 3.3962129697  EducationField\_Other 1.0247977557  EducationField\_Technical\_Degree 2.3911065654  EnvironmentSatisfaction 9.8298171497  Gender\_Female 2.8777790215  Gender\_Male 2.7837211447  HourlyRate 5.9247658465  JobInvolvement 8.2139560424  JobLevel 7.5322646424  JobRole\_Healthcare\_Representative 0.9617488864  JobRole\_Human\_Resources 0.7459471394  JobRole\_Laboratory\_Technician 3.6194024409  JobRole\_Manager 0.6393419249  JobRole\_Manufacturing\_Director 0.8459376795  JobRole\_Research\_Director 0.3857787109  JobRole\_Research\_Scientist 2.6430388636  JobRole\_Sales\_Executive 2.2851285134  JobRole\_Sales\_Representative 2.9547509506  JobSatisfaction 9.3200687502  MaritalStatus\_Divorced 2.1870320481  MaritalStatus\_Married 2.9745579132  MaritalStatus\_Single 5.1035919102  MonthlyIncome 6.0387502801  MonthlyRate 7.1840496708  NumCompaniesWorked 9.1508663658  OverTime\_No 10.4408604473  OverTime\_Yes 9.5534472611  PercentSalaryHike 6.2625343861  PerformanceRating 1.9269357522  RelationshipSatisfaction 8.2598181674  StockOptionLevel 7.6542745299  TotalWorkingYears 6.1617186125  TrainingTimesLastYear 8.4386111731  WorkLifeBalance 7.7289599303  YearsAtCompany 6.4262235992  YearsInCurrentRole 5.5724401372  YearsSinceLastPromotion 6.3969836987  YearsWithCurrManager 4.4410258734 |
| **Very significant variables:**  > ifelse(ImpVar > 8,"IMP",0)-> VeryImpVar  > VeryImpVar  IncNodePurity  Age "IMP"  BusinessTravel\_Non\_Travel "0"  BusinessTravel\_Travel\_Frequently "0"  BusinessTravel\_Travel\_Rarely "0"  DailyRate "0"  Department\_Human\_Resources "0"  Department\_Research\_And\_Development "0"  Department\_Sales "0"  DistanceFromHome "IMP"  Education "0"  EducationField\_Human\_Resources "0"  EducationField\_Life\_Sciences "0"  EducationField\_Marketing "0"  EducationField\_Medical "0"  EducationField\_Other "0"  EducationField\_Technical\_Degree "0"  EnvironmentSatisfaction "IMP"  Gender\_Female "0"  Gender\_Male "0"  HourlyRate "0"  JobInvolvement "IMP"  JobLevel "0"  JobRole\_Healthcare\_Representative "0"  JobRole\_Human\_Resources "0"  JobRole\_Laboratory\_Technician "0"  JobRole\_Manager "0"  JobRole\_Manufacturing\_Director "0"  JobRole\_Research\_Director "0"  JobRole\_Research\_Scientist "0"  JobRole\_Sales\_Executive "0"  JobRole\_Sales\_Representative "0"  JobSatisfaction "IMP"  MaritalStatus\_Divorced "0"  MaritalStatus\_Married "0"  MaritalStatus\_Single "0"  MonthlyIncome "0"  MonthlyRate "0"  NumCompaniesWorked "IMP"  OverTime\_No "IMP"  OverTime\_Yes "IMP"  PercentSalaryHike "0"  PerformanceRating "0"  RelationshipSatisfaction "IMP"  StockOptionLevel "IMP"  TotalWorkingYears "0"  TrainingTimesLastYear "IMP"  WorkLifeBalance "IMP"  YearsAtCompany "0"  YearsInCurrentRole "0"  YearsSinceLastPromotion "0"  YearsWithCurrManager "0" |

|  |
| --- |
| **XG Boost** |
| **XG boost Model building:**  >xgboost(data=xgtrainmatrix,label=xgtraindata$Attrition, nrounds=20, max\_depth=3,objective = "binary:logistic", eta=.03) -> xg  Here matrix of Traindata has been used, with “Binary:logistic” as we need the prediction of either 0 or 1  Optimal learning rate idendtified is 0.03 |
| > predict(xg,xgtestmatrix) -> predboost  > predboost  [1] 0.7249866128 0.2732670903 0.2732670903 0.2732670903 0.2732670903 0.2732670903 0.2732670903  [8] 0.2732670903 0.2732670903 0.7249866128 0.2732670903 0.7249866128 0.2732670903 0.7249866128  [15] 0.7249866128 0.2732670903 0.2732670903 0.2732670903 0.2732670903 0.2732670903 0.2732670903  [22] 0.2732670903 0.2732670903 0.2732670903 0.7249866128 0.2732670903 0.2732670903 0.2732670903  [29] 0.2732670903 0.2732670903 0.7249866128 0.2732670903 0.2732670903 0.7249866128 0.2732670903  [36] 0.2732670903 0.2732670903 0.2732670903 0.7249866128 0.2732670903 0.2732670903 0.2732670903  [43] 0.2732670903 0.2732670903 0.2732670903 0.7249866128 0.2732670903 0.7249866128 0.2732670903  [50] 0.7249866128 0.2732670903 0.2732670903 0.2732670903 0.2732670903 0.2732670903 0.2732670903  [57] 0.7249866128 0.2732670903 0.2732670903 0.2732670903 0.7249866128 0.2732670903 0.2732670903….. |
| > table(testdata$Attrition,pclass)  pclass  1 2  1 738 0  2 0 144  **Accuracy : 100%** |
|  |
|  |

**Model selection:**

Confusion Matrix gives a clear picture on selecting, which model to choose.

Considering only accuracy may not be a good practice as it may cover only the true positive and true negative and may fail practically with sensitivity and specificity measures.

We should have a trade-off between type1 and type 2 error. Accuracy tuning may result in more type1 and less type 2 error or vice versa. Some business cases can have the luxury of accommodating type2 error than type 1 error or vice versa.

Here with respect to our business case, when type 1 error is more, HR has to talk with more employees about their concerns.

If type 2 error is more, HR may miss to talk with those employees, which eventually results in Attrition.

Here we should focus to reduce type 2 error than type 1 error.

So we should select the model which gives more accuracy, with less type 2 error compared to type 1 error.

**Our Data Modeling Analysis:**

All the models we built, CART, NN and ensembling trechniques like RandomForest and Boosting are giving a significant accuracy of above 85%.

The top 5 influencing factors that HR may want to work shall be

* Overtime
* Job level
* Years at work
* Age
* JobRole
* Stockleveloptions

**Conclusion:**

On analyzing the various models and their accuracy in prediction, the **clear choice would be XGboost as it outperforms and gives 100% accuracy**. With the zero type 1 and type 2 error this is the unanimous winner among all models we tried like CART/ NN/ Random forest.