DDS\_401\_TeamNI\_Case\_Study\_2

Michael J Wolfe & Ayoade Dare

February 23, 2019

Career Attrition Analysis for DDS Talent Management

NaturalIntelligence Analytics

NaturalIntelligence Analytics would like to thank DDS Talent Management for the opportunity to explore job attrition data! The codebook below is intended to load, clean, and explore the data supplied to us for the purpose of discovering the top 3 factors that lead to attrition. Let’s get started!

## Let’s set up our environment and load the data!

suppressMessages(library(randomForest))  
suppressMessages(library(randomForestExplainer))  
suppressMessages(library(pscl))  
suppressMessages(library(RCurl))  
suppressMessages(library(ggplot2))  
suppressMessages(library(nnet))  
suppressMessages(library(kableExtra))  
suppressMessages(library(broom))  
jobURL <- getURL("https://raw.githubusercontent.com/mjwolfe91/DDS\_401\_TeamNI\_Case\_Study2/master/Data/CaseStudy2-data.csv")  
jobDF <- read.csv(text=jobURL, header=TRUE)  
#kable(head(jobDF), format = "html", align = "c") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))  
kable(head(jobDF), align = "c") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))

X.U.FEFF.Age

Attrition

BusinessTravel

DailyRate

Department

DistanceFromHome

Education

EducationField

EmployeeCount

EmployeeNumber

EnvironmentSatisfaction

Gender

HourlyRate

JobInvolvement

JobLevel

JobRole

JobSatisfaction

MaritalStatus

MonthlyIncome

MonthlyRate

NumCompaniesWorked

Over18

OverTime

PercentSalaryHike

PerformanceRating

RelationshipSatisfaction

StandardHours

StockOptionLevel

TotalWorkingYears

TrainingTimesLastYear

WorkLifeBalance

YearsAtCompany

YearsInCurrentRole

YearsSinceLastPromotion

YearsWithCurrManager

41

Yes

Travel\_Rarely

1102

Sales

1

2

Life Sciences

1

1

2

Female

94

3

2

Sales Executive

4

Single

5993

19479

8

Y

Yes

11

3

1

80

0

8

0

1

6

4

0

5

49

No

Travel\_Frequently

279

Research & Development

8

1

Life Sciences

1

2

3

Male

61

2

2

Research Scientist

2

Married

5130

24907

1

Y

No

23

4

4

80

1

10

3

3

10

7

1

7

37

Yes

Travel\_Rarely

1373

Research & Development

2

2

Other

1

4

4

Male

92

2

1

Laboratory Technician

3

Single

2090

2396

6

Y

Yes

15

3

2

80

0

7

3

3

0

0

0

0

33

No

Travel\_Frequently

1392

Research & Development

3

4

Life Sciences

1

5

4

Female

56

3

1

Research Scientist

3

Married

2909

23159

1

Y

Yes

11

3

3

80

0

8

3

3

8

7

3

0

27

No

Travel\_Rarely

591

Research & Development

2

1

Medical

1

7

1

Male

40

3

1

Laboratory Technician

2

Married

3468

16632

9

Y

No

12

3

4

80

1

6

3

3

2

2

2

2

32

No

Travel\_Frequently

1005

Research & Development

2

2

Life Sciences

1

8

4

Male

79

3

1

Laboratory Technician

4

Single

3068

11864

0

Y

No

13

3

3

80

0

8

2

2

7

7

3

6

## Hmmm, looks like there’s a little cleanup to do…let’s stratify some of the variables

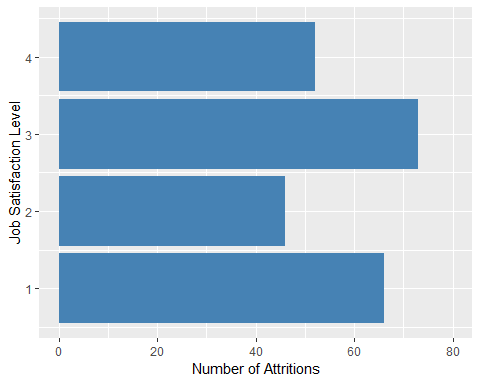
str(jobDF)

## 'data.frame': 1470 obs. of 35 variables:  
## $ X.U.FEFF.Age : int 41 49 37 33 27 32 59 30 38 36 ...  
## $ Attrition : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 1 1 1 ...  
## $ BusinessTravel : Factor w/ 3 levels "Non-Travel","Travel\_Frequently",..: 3 2 3 2 3 2 3 3 2 3 ...  
## $ DailyRate : int 1102 279 1373 1392 591 1005 1324 1358 216 1299 ...  
## $ Department : Factor w/ 3 levels "Human Resources",..: 3 2 2 2 2 2 2 2 2 2 ...  
## $ DistanceFromHome : int 1 8 2 3 2 2 3 24 23 27 ...  
## $ Education : int 2 1 2 4 1 2 3 1 3 3 ...  
## $ EducationField : Factor w/ 6 levels "Human Resources",..: 2 2 5 2 4 2 4 2 2 4 ...  
## $ EmployeeCount : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ EmployeeNumber : int 1 2 4 5 7 8 10 11 12 13 ...  
## $ EnvironmentSatisfaction : int 2 3 4 4 1 4 3 4 4 3 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 1 2 2 1 2 2 1 2 2 2 ...  
## $ HourlyRate : int 94 61 92 56 40 79 81 67 44 94 ...  
## $ JobInvolvement : int 3 2 2 3 3 3 4 3 2 3 ...  
## $ JobLevel : int 2 2 1 1 1 1 1 1 3 2 ...  
## $ JobRole : Factor w/ 9 levels "Healthcare Representative",..: 8 7 3 7 3 3 3 3 5 1 ...  
## $ JobSatisfaction : int 4 2 3 3 2 4 1 3 3 3 ...  
## $ MaritalStatus : Factor w/ 3 levels "Divorced","Married",..: 3 2 3 2 2 3 2 1 3 2 ...  
## $ MonthlyIncome : int 5993 5130 2090 2909 3468 3068 2670 2693 9526 5237 ...  
## $ MonthlyRate : int 19479 24907 2396 23159 16632 11864 9964 13335 8787 16577 ...  
## $ NumCompaniesWorked : int 8 1 6 1 9 0 4 1 0 6 ...  
## $ Over18 : Factor w/ 1 level "Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ OverTime : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 1 2 1 1 1 ...  
## $ PercentSalaryHike : int 11 23 15 11 12 13 20 22 21 13 ...  
## $ PerformanceRating : int 3 4 3 3 3 3 4 4 4 3 ...  
## $ RelationshipSatisfaction: int 1 4 2 3 4 3 1 2 2 2 ...  
## $ StandardHours : int 80 80 80 80 80 80 80 80 80 80 ...  
## $ StockOptionLevel : int 0 1 0 0 1 0 3 1 0 2 ...  
## $ TotalWorkingYears : int 8 10 7 8 6 8 12 1 10 17 ...  
## $ TrainingTimesLastYear : int 0 3 3 3 3 2 3 2 2 3 ...  
## $ WorkLifeBalance : int 1 3 3 3 3 2 2 3 3 2 ...  
## $ YearsAtCompany : int 6 10 0 8 2 7 1 1 9 7 ...  
## $ YearsInCurrentRole : int 4 7 0 7 2 7 0 0 7 7 ...  
## $ YearsSinceLastPromotion : int 0 1 0 3 2 3 0 0 1 7 ...  
## $ YearsWithCurrManager : int 5 7 0 0 2 6 0 0 8 7 ...

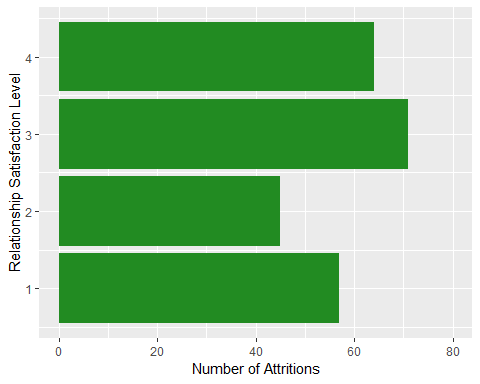
jobDF$AgeGroup <- cut(jobDF$X.U.FEFF.Age, c(-Inf, 20, 29, 39, 49, 59, Inf))  
levels(jobDF$AgeGroup) <- c("<20", "20-29", "30-39", "40-49", "50-59", "60+")  
jobDF$GenderInd <- 0  
jobDF[jobDF$Gender == "Male", ]$GenderInd <- 1  
drops <- c("X.U.FEFF.Age","Gender")  
jobDF <- jobDF[ , !(names(jobDF) %in% drops)]

## Let’s do some exploratory data analysis

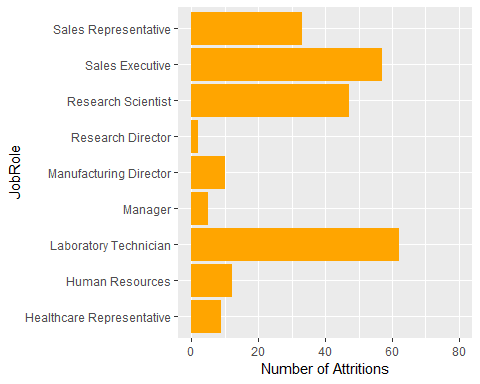
jobDF.eda <- jobDF  
jobDF.eda$AttritionInd <- 0  
jobDF.eda[jobDF.eda$Attrition == "Yes", ]$AttritionInd <- 1  
JSsum <- aggregate(jobDF.eda$AttritionInd, by=list(Category=jobDF.eda$JobSatisfaction), FUN=sum)  
ggplot(JSsum, aes(y=x, x=Category)) + geom\_bar(position="dodge", stat="identity", fill=c("steelblue")) + scale\_x\_continuous(name="Job Satisfaction Level") + scale\_y\_continuous(name="Number of Attritions", limits=c(0, 80)) + coord\_flip()



RelSum <- aggregate(jobDF.eda$AttritionInd, by=list(Category=jobDF.eda$RelationshipSatisfaction), FUN=sum)  
ggplot(RelSum, aes(y=x, x=Category)) + geom\_bar(position="dodge", stat="identity", fill=c("forestgreen")) + scale\_x\_continuous(name="Relationship Satisfaction Level") + scale\_y\_continuous(name="Number of Attritions", limits=c(0, 80)) + coord\_flip()



RoleSum <- aggregate(jobDF.eda$AttritionInd, by=list(JobRole=jobDF.eda$JobRole), FUN=sum)  
ggplot(RoleSum, aes(y=x, x=JobRole)) + geom\_bar(position="dodge", stat="identity", fill=c("orange")) + scale\_y\_continuous(name="Number of Attritions", limits=c(0, 80)) + coord\_flip()



## Not sure where to begin…so let’s test them all!

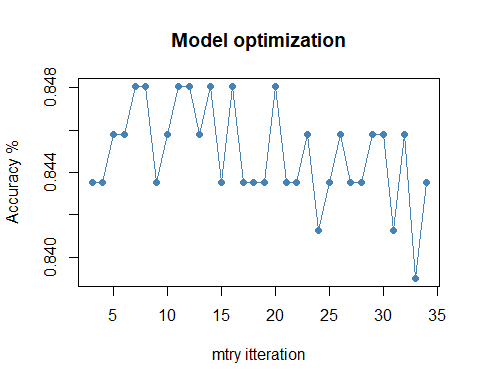
Since we are looking for the top 3 variables out of several possibilities, we will use a machine learning technique known as a random forest. A random forest is a panel of decision trees designed to test several permutations of variable interactions to see which variables seem to have the most influence on a particular outcome. This makes it a powerful technique in exploration. The first step is to build a training set & test set. The training set is 70% of the data, while the test set is 30%.

set.seed(26)  
train <- sample(nrow(jobDF),0.7\*nrow(jobDF), replace = FALSE)  
jobDF.train <- jobDF[train,]  
jobDF.test <- jobDF[-train,]  
jobs.rf <- randomForest(Attrition ~ ., data=jobDF.train, importance=TRUE)  
print(jobs.rf)

##   
## Call:  
## randomForest(formula = Attrition ~ ., data = jobDF.train, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 5  
##   
## OOB estimate of error rate: 13.41%  
## Confusion matrix:  
## No Yes class.error  
## No 862 4 0.004618938  
## Yes 134 29 0.822085890

## Identify the optimal mtry

a=c()  
i=5  
for (i in 3:34) {  
 jobs.rf2<-randomForest(Attrition ~.,data = jobDF.train, ntree=500, mtry=i, importance=TRUE)  
 pred\_test\_jobs<-predict(jobs.rf2,jobDF.test,type ="class")  
 a[i-2]=mean(pred\_test\_jobs==jobDF.test$Attrition)  
}  
plot(3:34,a,pch=19,col="steel blue",type = "o",xlab = "mtry itteration", ylab = "Accuracy %",main = "Model optimization")



## Looks like 24 mtry’s generates the most accurate model

jobs.rf3 <- randomForest(Attrition ~ ., data=jobDF.train, ntree=500, importance=TRUE, mtry=24)  
print(jobs.rf3)

##   
## Call:  
## randomForest(formula = Attrition ~ ., data = jobDF.train, ntree = 500, importance = TRUE, mtry = 24)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 24  
##   
## OOB estimate of error rate: 12.83%  
## Confusion matrix:  
## No Yes class.error  
## No 853 13 0.01501155  
## Yes 119 44 0.73006135

## Let’s predict!

jobRFPred <-predict(jobs.rf3,jobDF.test,type = "class")  
mean(pred\_test\_jobs==jobDF.test$Attrition)

## [1] 0.8435374

#kable(table(pred\_test\_jobs,jobDF.test$Attrition), format = "html", align = "c") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))  
kable(table(pred\_test\_jobs,jobDF.test$Attrition), align = "c") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))

No

Yes

No

361

63

Yes

6

11

## Prediction test supports the accuracy we found earlier. Let’s use this to pick our top 3 variables.

importanceDF <- data.frame(round(importance(jobs.rf3),2))  
sortYes <- importanceDF[order(importanceDF$Yes,decreasing = TRUE),]  
#kable(sortYes, format = "html", align = "c") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))  
kable(sortYes, align = "c") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))

No

Yes

MeanDecreaseAccuracy

MeanDecreaseGini

OverTime

18.83

32.14

30.70

19.90

MonthlyIncome

11.97

15.21

19.07

26.81

JobLevel

4.10

7.53

7.85

3.36

TotalWorkingYears

11.36

7.28

13.80

12.34

JobInvolvement

5.37

7.08

7.93

7.84

DailyRate

5.51

6.92

8.54

19.36

EnvironmentSatisfaction

4.69

6.19

7.21

8.98

JobRole

10.84

4.69

12.02

14.13

YearsAtCompany

8.61

4.59

10.42

11.17

YearsWithCurrManager

8.70

4.24

9.51

7.60

AgeGroup

9.35

4.02

10.19

8.30

BusinessTravel

4.77

3.56

5.52

4.14

YearsInCurrentRole

7.34

3.48

8.66

5.04

EducationField

3.12

3.44

4.10

10.51

DistanceFromHome

1.59

3.29

2.82

11.65

JobSatisfaction

1.67

2.93

2.76

6.07

YearsSinceLastPromotion

1.88

2.67

2.80

6.70

StockOptionLevel

5.41

2.54

6.03

6.09

MaritalStatus

5.51

2.49

5.59

4.57

Department

-0.46

2.46

0.50

1.06

NumCompaniesWorked

3.34

1.82

3.76

9.79

WorkLifeBalance

-0.13

1.44

0.46

6.61

TrainingTimesLastYear

3.08

1.12

3.23

7.86

HourlyRate

0.66

0.95

1.08

12.62

EmployeeCount

0.00

0.00

0.00

0.00

Over18

0.00

0.00

0.00

0.00

StandardHours

0.00

0.00

0.00

0.00

GenderInd

-0.09

-0.30

-0.32

1.58

PerformanceRating

-0.49

-0.85

-0.77

0.59

EmployeeNumber

0.25

-1.04

-0.26

10.93

Education

0.65

-1.18

0.10

4.86

PercentSalaryHike

0.82

-1.43

0.10

6.41

RelationshipSatisfaction

0.23

-1.55

-0.41

4.64

MonthlyRate

-2.64

-1.98

-3.33

13.05

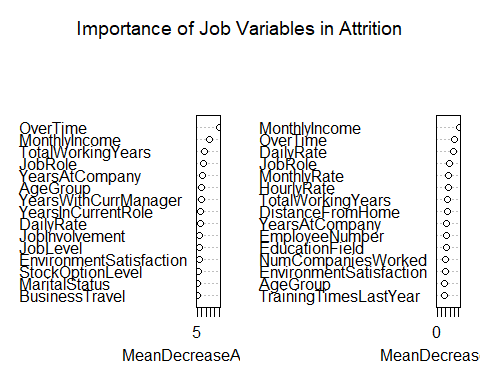
According to our random forest analysis, the top 3 factors that contribute to attrition are Overtime, Monthly Income, and Environment Satisfaction, with ~84% accuracy.

## Let’s take a look at a more detailed breakdown

#explain\_forest(jobs.rf3, data=jobDF)

## Let’s plot importance for each variable

varImpPlot(jobs.rf3, main="Importance of Job Variables in Attrition", n.var=15)



It seems clear that the 3 factors we highlighted earlier are our best bet, according to this algorithm. Let’s test!

## Now that we have the top 3, let’s test the model!

Since this is a model with a binary response (someone will attrite or they will not), we will test a logistic regression model. It’s important to recognize that this model predicts if the included variables will have an impact (either way) on the attrition outcome, not that they are entirely predictive of positive attrition.

jobs.glm <- glm(Attrition ~ MonthlyIncome + EnvironmentSatisfaction + OverTime, data=jobDF, family=binomial(link='logit'))  
#kable(tidy(jobs.glm), format = "html", align = "c") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))  
#kable(pR2(jobs.glm), format = "html", align = "c") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))  
kable(tidy(jobs.glm), align = "c") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))

term

estimate

std.error

statistic

p.value

(Intercept)

-0.5231823

0.2259298

-2.315685

0.0205755

MonthlyIncome

-0.0001402

0.0000227

-6.185634

0.0000000

EnvironmentSatisfaction

-0.3396194

0.0693113

-4.899914

0.0000010

OverTimeYes

1.4741367

0.1538107

9.584095

0.0000000

kable(pR2(jobs.glm), align = "c") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))

x

llh

-571.7344869

llhNull

-649.2913504

G2

155.1137270

McFadden

0.1194485

r2ML

0.1001431

r2CU

0.1707119

In a linear regression model, the adjusted R-Squared is the most broadly accepted statistic for “accuracy.” In logistic regression, the closest equivalent is the McFadden log likelihood. In this case, the model with the above parameters produces 11.9% McFadden score - indicating that this model contributes to roughly 11.9% of the variance in attrition outcomes.

## Other observations

Our random forest algorithm has also postulated the top factors that contribution to a person remaining in their job. Let’s go through a similar exercise:

sortNo <- importanceDF[order(importanceDF$No,decreasing = TRUE),]  
#kable(sortNo, format = "html", align = "c") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))  
kable(sortNo, align = "c") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))

No

Yes

MeanDecreaseAccuracy

MeanDecreaseGini

OverTime

18.83

32.14

30.70

19.90

MonthlyIncome

11.97

15.21

19.07

26.81

TotalWorkingYears

11.36

7.28

13.80

12.34

JobRole

10.84

4.69

12.02

14.13

AgeGroup

9.35

4.02

10.19

8.30

YearsWithCurrManager

8.70

4.24

9.51

7.60

YearsAtCompany

8.61

4.59

10.42

11.17

YearsInCurrentRole

7.34

3.48

8.66

5.04

DailyRate

5.51

6.92

8.54

19.36

MaritalStatus

5.51

2.49

5.59

4.57

StockOptionLevel

5.41

2.54

6.03

6.09

JobInvolvement

5.37

7.08

7.93

7.84

BusinessTravel

4.77

3.56

5.52

4.14

EnvironmentSatisfaction

4.69

6.19

7.21

8.98

JobLevel

4.10

7.53

7.85

3.36

NumCompaniesWorked

3.34

1.82

3.76

9.79

EducationField

3.12

3.44

4.10

10.51

TrainingTimesLastYear

3.08

1.12

3.23

7.86

YearsSinceLastPromotion

1.88

2.67

2.80

6.70

JobSatisfaction

1.67

2.93

2.76

6.07

DistanceFromHome

1.59

3.29

2.82

11.65

PercentSalaryHike

0.82

-1.43

0.10

6.41

HourlyRate

0.66

0.95

1.08

12.62

Education

0.65

-1.18

0.10

4.86

EmployeeNumber

0.25

-1.04

-0.26

10.93

RelationshipSatisfaction

0.23

-1.55

-0.41

4.64

EmployeeCount

0.00

0.00

0.00

0.00

Over18

0.00

0.00

0.00

0.00

StandardHours

0.00

0.00

0.00

0.00

GenderInd

-0.09

-0.30

-0.32

1.58

WorkLifeBalance

-0.13

1.44

0.46

6.61

Department

-0.46

2.46

0.50

1.06

PerformanceRating

-0.49

-0.85

-0.77

0.59

MonthlyRate

-2.64

-1.98

-3.33

13.05

It looks like some of the same factors that lead to attrition can also contribute to someone remaining in their current role. Let’s adjust the model for a “No” response assumption:

jobsNo.glm <- glm(Attrition ~ MonthlyIncome + TotalWorkingYears + OverTime, data=jobDF, family=binomial(link='logit'))  
#kable(tidy(jobsNo.glm), format = "html", align = "c") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))  
#kable(pR2(jobsNo.glm), format = "html", align = "c") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))  
kable(tidy(jobsNo.glm), align = "c") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))

term

estimate

std.error

statistic

p.value

(Intercept)

-1.2576143

0.1533876

-8.198929

0.0000000

MonthlyIncome

-0.0000681

0.0000309

-2.204935

0.0274586

TotalWorkingYears

-0.0540751

0.0174504

-3.098782

0.0019432

OverTimeYes

1.3964687

0.1508408

9.257896

0.0000000

kable(pR2(jobsNo.glm), align = "c") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))

x

llh

-578.7809665

llhNull

-649.2913504

G2

141.0207678

McFadden

0.1085959

r2ML

0.0914747

r2CU

0.1559349

This produces a McFadden score of 10.8%.

## Conclusion

While on the surface, it appears the variables we selected using the random forest do not have that large of an impact, this does not mean they are not accurate. There are almost 20 possible variables and many different permutations therein, meaning true “accuracy” cannot be measured without training and testing the model. The next steps would be to obtain more data.