Employee Attrition Modelling: Part 1

SFL Scientific

August 2017

The Data Problem

Understanding employee turnover is one of the most important aspects for any HR department. Employee attrition is defined as the rate at which employees leave a company.

The goal of this review series is to understand employee attrition and determine the most dominant contributing factors that govern this turnover for an IBM dataset. This dataset includes employees' attrition information and other basic employees' information which we will explore detaily later. The benefits to the company are substantial: not only is the best talent retained, but recruitment and training costs may be reduced significantly as well.

Through this kind of analysis, we can understand how many employees are likely to leave, while also determining which employees are at the highest risk and for what reasons. In this analysis, you will see what factors are most significant to employee attrition shown by models. Companies face significant costs for having to search for, interview, and hire new employees. In general therefore, a company is highly motivated to retain their employees for a significant period of time.

The analysis will be split into four separate blogs:

1. An Exploratory Data Analysis (EDA), where we will take a first look at the dataset,
2. The Data Modelling with Machine Learning
3. Discussion of most important features
4. A look into the RShiny App

The readers who have already gotten knowledges in the IBM dataset can skip the first blog and go to the second blog directly and of course, you are encouraged to read the whole since we may use some other visual tools than yours. For those readers who are not comfortable with machine learning algorithms, you can skip the second blog and take a glance at the discussion in the third blog. Don't forget to have a try with our Shiny app in the last blog.

Before we start exploring the dataset, let us open it and do some data cleaning.

Getting data

```
# download the data from
# https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset
# save it to the current rmd working directory
d<-read.csv("WA_Fn-UseC_-HR-Employee-Attrition.csv") # read the csv file

# load packages
library(ggplot2)
library(gridExtra)
```

```r
library(ggthemes)
library(corrplot)
```

Data cleaning

```r
# rename the first column
colnames(d)[1] <-  "Age"
par(xpd=TRUE)

# change categorical to numerical
# only numatical values support the calculation of
# correlation matrix for the columns
d$Attrition <-  as.integer(as.factor(d$Attrition))-1
d$BusinessTravel <-  as.integer(as.factor(d$BusinessTravel))
d$Department <-  as.integer(as.factor(d$Department))
d$JobRole <-  as.integer(as.factor(d$JobRole))
d$MaritalStatus <-  as.integer(as.factor(d$MaritalStatus))
d$OverTime <-  as.integer(as.factor(d$OverTime))
d$EducationField <-  as.integer(as.factor(d$EducationField))
d$Gender <- as.integer(as.factor(d$Gender))

# delete unwanted columns
# they only have a unique value or
# they are not related in our case
d$StandardHours <- NULL
d$PerformanceRating <- NULL
d$Over18 <- NULL
d$EmployeeCount <- NULL
d$JobLevel <- NULL
d$DailyRate <- NULL
d$HourlyRate <- NULL
d$DailyRate <- NULL
d$MonthlyRate <- NULL
d$PercentSalaryHike <- NULL
```

Part 1: Exploratory Data Analysis

The dataset contains several pieces of information about each employee such as their department, job satisfaction, years at company, work/life balance and so on. Of all these, there are five that can be used to subset the users in our Shiny app: age, gender, education level, monthly income and marital status. Note that we do not recommend subsetting the dataset by more than three parameters as there may be no data satisfying.

In terms of a machine learning analysis, the data has to be initially cleaned before it can be used. For most projects, cleaning the data is often the most time consuming aspect of the entire process. Generally you would want to fill in missing values, understand (potentially discard) outliers, correct erroneous ones, fix formatting issues and standardize categories. The goal is to make the data as consistent and relevant across the board as possible. This will allow for the maximum accuracy of the final model.

We first conduct an exploratory analysis on the dataset using visual tools, which allows us to summarize the

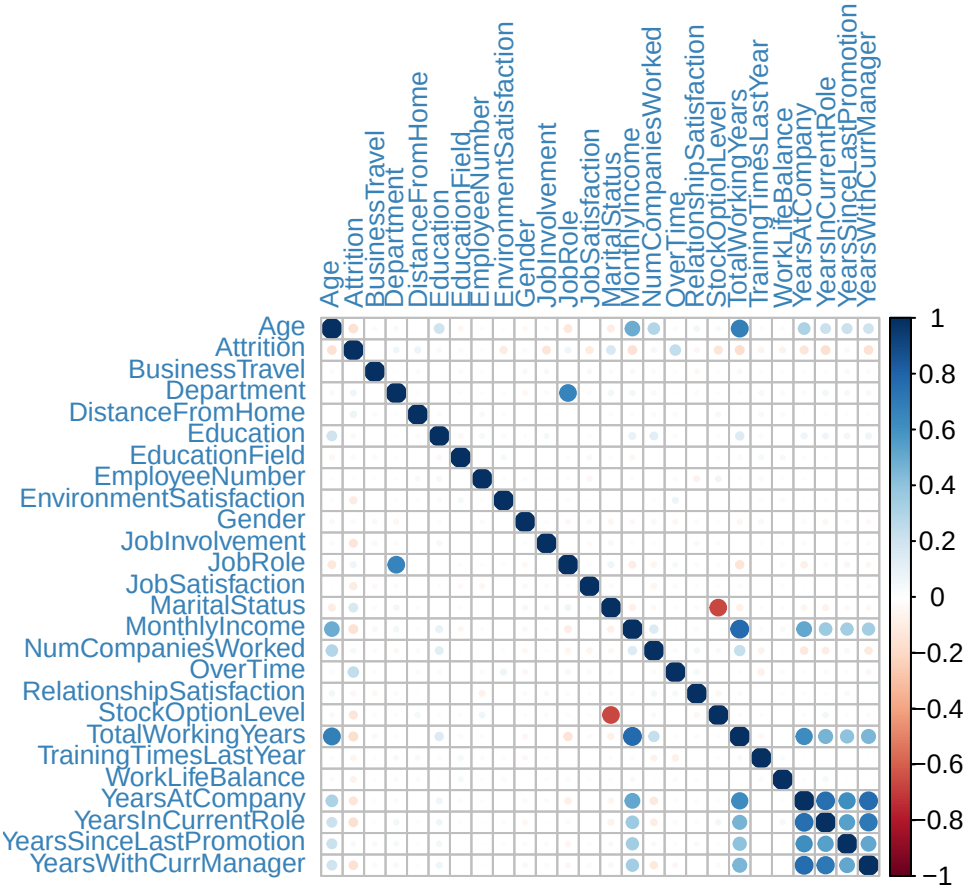

Figure 1: Correlation Matrix for all features. A larger dot indicates that the correlation between these selected features is stronger, whereas the color denotes the strength of the positive (blue) or negative (red) correlation coefficient.

main characteristics of a dataset. From here, we perform machine learning modelling that will determine the probability that each individual will attrite, thus, uncovering the most important factors that lead to overall employee turnover. Based on the needs of the employer, this analysis can also be narrowed down to determine key factors governing attrition for particular demographics, job titles, working groups, and indeed specific individuals.

The Correlation Matrix and EDA Plots

```
# set the margin
par(xpd=TRUE)

# draw the correlation matrix plot
corrplot(cor(d), method="circle", tl.col="#3982B7",mar = c(2, 0, 0, 0),tl.cex = 0.8)
```

The correlation matrix in Figure 1 displays the linear correlation between every pair of features in the form of dots of varying colors and sizes. When two variables are correlated, for example Age and TotalWorkingYears, we are essentially observing that change in one variable is accompanied by change in the other.

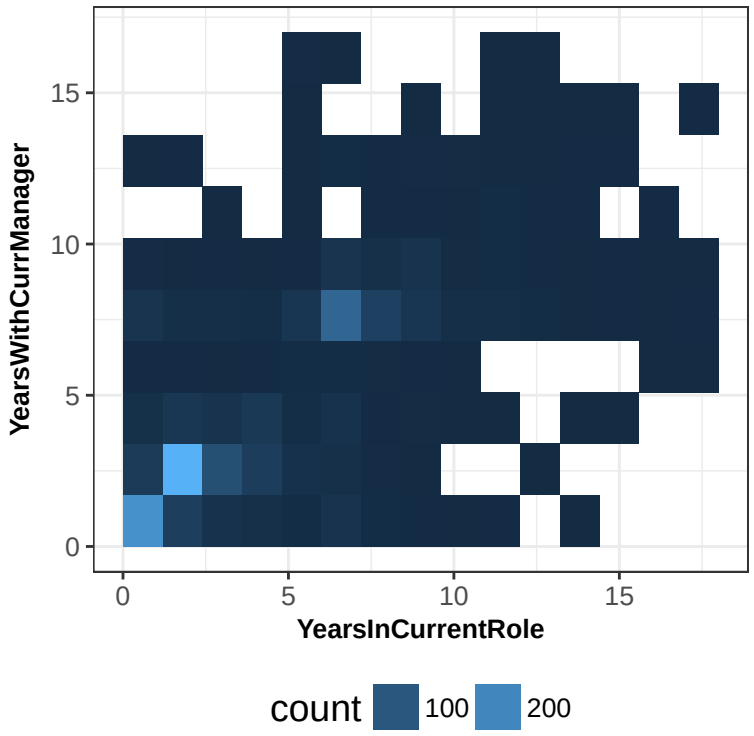

Figure 2: Correlation between Years With Current Manager and Years At Company. Figure shows the relatively high correlation between the two features YearsInCurrentRole and YearsWithCurrManage; this relationship seems intuitive due to the obvious inter-related nature of these features.

In this large matrix, it is clear that the majority of features are uncorrelated. However, even for those variables that are correlated, care must be taken when interpreting the correlation as it does not necessarily imply a causal relationship.

```
# construct correlation plot using ggplot2 stat_bin2d
ggplot(d, aes(YearsInCurrentRole, YearsWithCurrManager))+              #set axis
stat_bin2d(bins = c(15, 10))+                                          #set bin numbers
guides(colour = guide_legend(override.aes = list(alpha = 1)),          #set theme and legend
fill = guide_legend(override.aes = list(alpha = 1)))+
theme_bw()+theme(axis.text=element_text(size=10),
axis.title=element_text(size=10,face="bold"),
legend.text=element_text(size=10),legend.title=element_text(size=14),
legend.position = "bottom")+
xlab("YearsInCurrentRole")+ylab("YearsWithCurrManager")                #rename labels
```

This application has an additional functionality: by clicking any element in the correlation matrix, a 2D histogram is displayed in order to better observe the correlation between those features as shown in Figure 2. Correlation between variables allows us to determine the overlap and redundancies between features in the dataset. In general, the machine learning algorithm should be given as much uncorrelated information as possible to maximise the predictive accuracy and model interpretability.

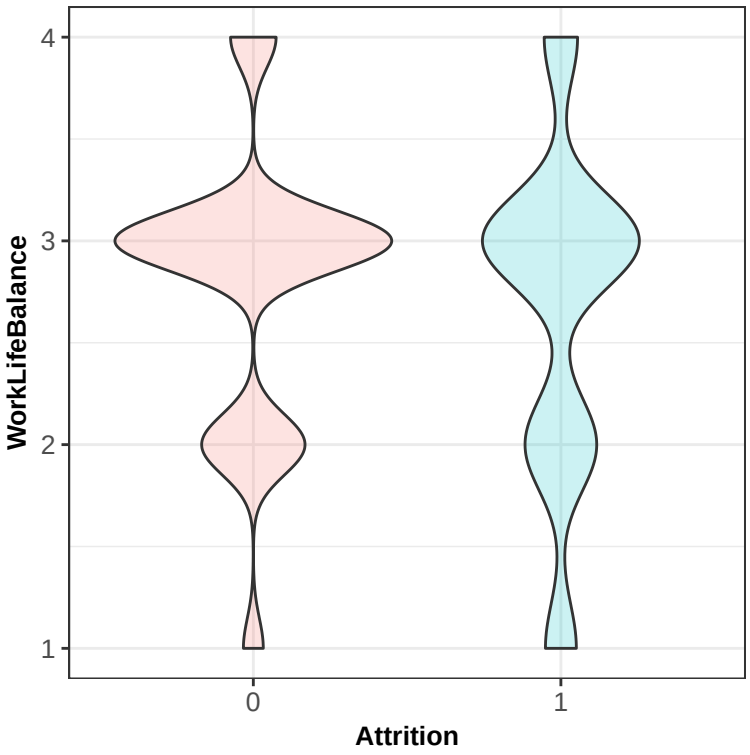

Figure 3: Violin plots of Work/Life Balance, separated by attrition. Figure shows the violin plots for the WorkLifeBalance variable with those that attrite tending to have fewer mid-range scores.

```
# construct violin plot using ggplot2 geom_violin
ggplot(d, aes(factor(Attrition), WorkLifeBalance))+                    #set axis
geom_violin(alpha = 0.2, aes(fill = factor(Attrition)))+               #set violin plot
theme_bw()+                                                            #set theme and legend
guides(fill=FALSE)+theme(axis.text=element_text(size=10),
axis.title=element_text(size=10,face="bold"),
legend.text=element_text(size=10),
legend.title=element_text(size=14),legend.position = "bottom")+
xlab("Attrition")                                                      #rename label
```

Alternatively, clicking the elements along the leading diagonal will output violin plots of the selected features, bucketed by the true underlying attrition value (1 indicating employees that attrite, and 0 indicated those that remain). Figure 3 shows the violin plots for the WorkLifeBalance variable with those that attrite tending to have fewer mid-range scores.

Unlike box plots, violin plots show the full distribution of the data, which is particularly useful if the data is multimodal. Such plots show the differences in employees that attrite and those that do not. Further, these plots show a first indication of the feature's importance to the machine learning model to predict attrition.

```
# factorize Education column
d$Education <-  as.factor(d$Education)
d$Gender <-  ifelse(d$Gender == 1, "female","male")
d$Gender <-  as.factor(d$Gender)

# construct plots using geom_boxplot
Age_plot1 <- ggplot(d, aes(as.factor(Attrition),Age,fill=Education)) + #set axis
```

```
              geom_boxplot(width = 0.5) +                                    #set boxplot
              theme(axis.text=element_text(size=10),                         #set theme and legend
              axis.title=element_text(size=10,face="bold"),
              legend.text=element_text(size=10),
              legend.title=element_text(size=10)) +
              theme_hc()+ scale_colour_hc()+
              xlab("Attrition")                                             #rename label


Age_plot2 <- ggplot(d, aes(as.factor(Attrition),Age,fill=Gender)) +
              geom_boxplot(width = 0.4) +
              theme(axis.text=element_text(size=10),
              axis.title=element_text(size=10,face="bold"),
              legend.text=element_text(size=10),
              legend.title=element_text(size=14)) +
              theme_hc()+ scale_colour_hc()+
              xlab("Attrition")


Income_plot1 <- ggplot(d,aes(as.factor(Attrition),MonthlyIncome,
                 fill=Education)) +
                 geom_boxplot(width =  0.5)+theme(axis.text=element_text(size=10),
                 axis.title=element_text(size=10,face="bold"),
                 legend.text=element_text(size=10),
                 legend.title=element_text(size=14)) +
                 theme_hc()+ scale_colour_hc()+
                 xlab("Attrition")


Income_plot2 <- ggplot(d, aes(as.factor(Attrition),MonthlyIncome,
                 fill=Gender)) +
                 geom_boxplot(width = 0.4) +   theme(axis.text=element_text(size=10),
                 legend.text=element_text(size=10),
                 legend.title=element_text(size=14)) +
                 theme_hc()+ scale_colour_hc()+
                 xlab("Attrition")

d$Gender <- as.integer(d$Gender)

# grid the plots together
grid.arrange(Age_plot1,Income_plot1,Age_plot2,Income_plot2,ncol = 2)
```

Those boxplots in Figure4 show us a clear bivariate relationship between attrition and age, monthly income, given different education levels or gender. In general, attrites tend to be younger and their education levels as well as monthly income are lower than non-attrites. Monthly income is a key factor to attrition. However, company could not give employees as much salary as they want so increasing monthly income is not pracitcal to the majority company. There are some outliers indicating by the plots, for example, even in the attrition set, there are still some employees whose monthly income is very high.

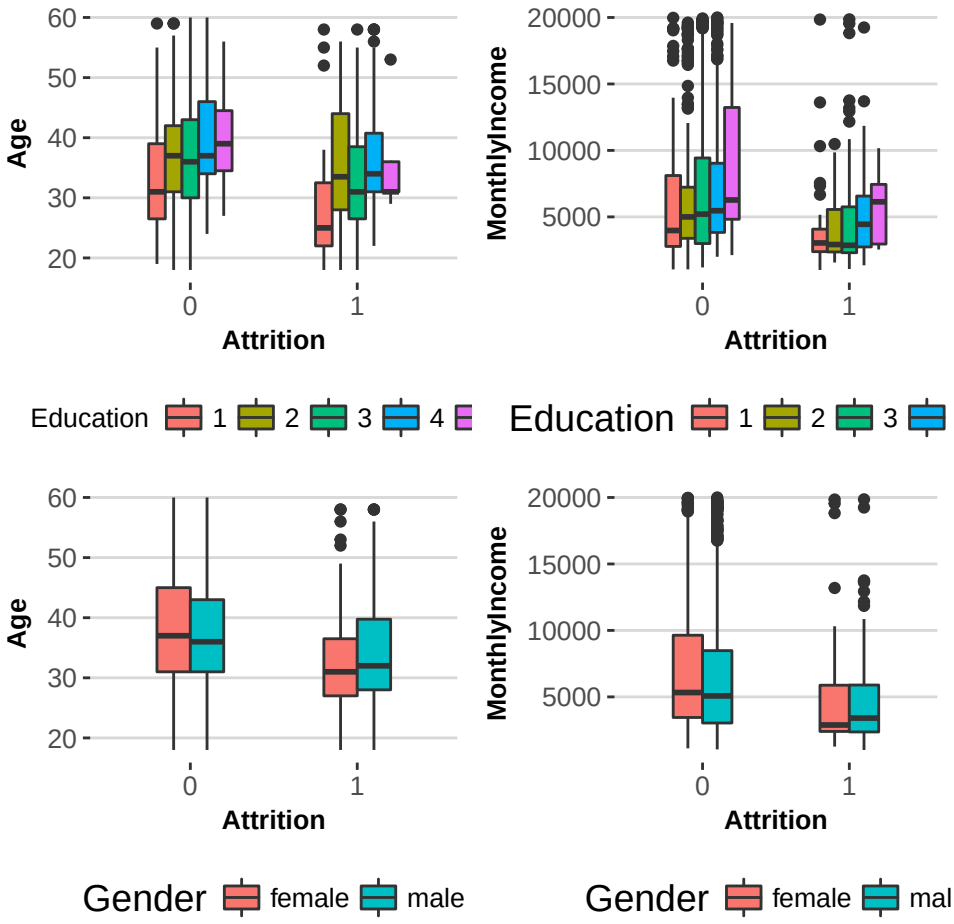

Figure 4: Boxplots of Age and MonthlyIncome, separated by attrition, gridded by education or gender. Figure shows that attrite tends to have lower MonthlyIncome, lower Education level and tends to be younger.

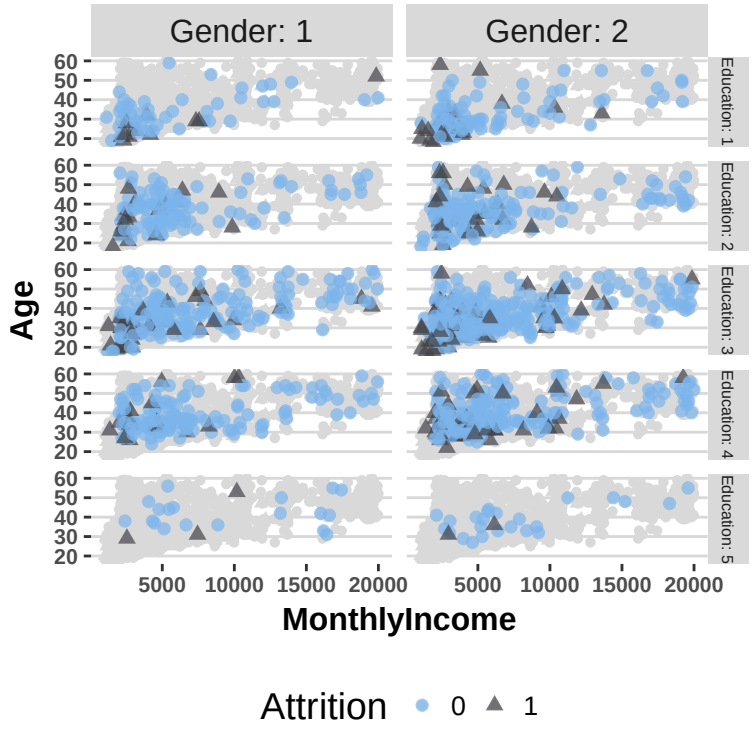

Figure 5: points plot of Age and MonthlyIncome, separated by attrition, gridded by education or gender.

```
# factorize attrition
d$Attrition = as.factor(d$Attrition)

# construct plot using geom_point
ggplot(d, aes(MonthlyIncome,Age, color=Attrition,shape=Attrition))+   #set axis
geom_point(data = transform(d, Education = NULL, Gender = NULL),      #set point plot
colour = "grey85") + geom_point(size = 2,alpha = 0.7)+
facet_grid(Education~Gender,labeller = "label_both") +                #set facet grid
theme(axis.text=element_text(size=8,face="bold"),                     #set theme and label
axis.title=element_text(size=12,face="bold"),
strip.text.x = element_text(size = 12),
strip.text.y = element_text(size = 6),
legend.text=element_text(size=10),
legend.title=element_text(size=14))+
theme_hc()+ scale_colour_hc()
```

More detaily, in Figure5, we subset the employees by gender and education level. The plot shows us a pretty much same result than the previous one. Additionally, we can see the number of employees in each subsets vary a lot. The number of attrites is much smaller than the number of non-attrites which is a usual company scenario. Here is the reason again why we do not recommand subsetting the dataset by more than three parameters.

We have finished the part1 and our Exploratory data analysis present us some basic information about the dataset. Last but not least, save our workspace so that we can directly get the cleaned dataset in later parts.

```r
# save part2 workspace
save.image("part1.RData")
```

Remarks

This type of exploratory data analysis allows us to visualise how the data looks, spot any outliers, potential issues where the model will perform poorly and which features might yield large predictive power.

For more details on this or any potential analyses, please visit us at http://sflscientific.com or contact mluk@sflscientific.com.