### **Employee Retention with R Based Data Science Accelerator**

by Le Zhang (Data Scientist, Microsoft) and Graham Williams (Director of Data Science, Microsoft)

Employee retention has been and will continue to be one of the biggest challenges of a company. While classical tactics such as promotion, competitive perks, etc. are practiced as ways to retain employees, it is now a hot trend to rely on machine learning technology to discover behavioral patterns with which a company can understand their employees better.

[](http://revolution-computing.typepad.com/.a/6a010534b1db25970b01b7c8dba088970b-pi)

Employee demographic data has been studied and used for analyzing [employees’ inclination towards leaving a company](http://amj.aom.org/content/44/6/1102.abstract). Nowadays, with the proliferation of the Internet, employees’ behavior can better be understood and analyzed through such data as internal and external social media postings. Such data can be leveraged for the analysis of, for example, sentiment and thereby [determination of an employees likelihood of leaving the company](http://www.typepad.com/site/blogs/6a010534b1db25970b010536040741970b/compose/preview/%5b2%5dhttps:/hbr.org/2016/01/sentiment-analysis-can-do-more-than-prevent-fraud-and-turnover). Novel cognitive computing technology based on artificial intelligence tools empower today’s HR departments to identify staff who are likely to churn before they do. Through pro-active intervention HR can better manage staff to encourage them to remain longer term with the company.

This blog post introduces an R based data science accelerator that can be quickly adopted by a data scientist to prototype a [solution for the employee attrition prediction scenario](https://github.com/Microsoft/acceleratoRs/tree/master/EmployeeAttritionPrediction).  The prediction is based on two types of employee data that are typically already collected by companies:

1. Static data which does not tend to change over time. This type of data may refer to demographic and organizational data such as age, gender, title, etc. A characteristic of this type of data is that within a certain period they do not change or solely change in a deterministic way. For instance, years of service of an employee is static as the number increments every year.
2. The second type of data is the dynamically evolving information about an employee. Recent studies revealed that sentiment is playing a critical role in employee attrition prediction. Classical measures of sentiment require employee surveys of work satisfaction work. Social media posts become useful for sentiment analysis as employees may express their feelings about work. Non-structural data such as text can be collected for mining patterns which are indicative of employees with different inclinations for churn.

Attrition prediction is a scenario that takes historic employee data as input to then identify individuals that are inclined to leave. The basic procedure is to extract features from the available data that might have previously been manually analyzed and to build predictive models based on a training set with labels relating to the employment status. Normally it can be formalized as a supervised classification problem, while the uniqueness is that population of employees with different employment status may not be equal. Training such an imbalanced data set requires resampling or cost-sensitive learning techniques. For sentiment analysis on unstructured data such as text, pre-processing techniques that extract analysis-friendly quantitative features should be applied. Commonly used feature extraction methods for text analysis include word-to-vector, term frequency, or term frequency and inverse document frequency, etc. Algorithms for building the model depend on the data characteristics. In case a specific algorithm does not yield the desired results, we have found that ensemble techniques can be deployed to effectively boost model performance.

[](http://revolution-computing.typepad.com/.a/6a010534b1db25970b01b7c8dba095970b-pi)

The data science language R is a convenient tool for performing HR churn prediction analysis. A lightweight data science accelerator that demonstrates the process of predicting employee attrition is shared in the [Github](https://github.com/Microsoft/acceleratoRs/tree/master/EmployeeAttritionPrediction). The walk-through basically shows cutting-edge machine learning and text mining techniques applied in R.

* Taking the data set from <https://community.watsonanalytics.com/wp-content/uploads/2015/03/WA_Fn-UseC_-HR-Employee-Attrition.xlsx> for example, one can easily visualize and perform correlation analysis with R packages. For instance, we may plot the distribution of monthly income of employees from different departments and intuitively analyze whether income can be a factor that affects attrition.

library(ggplot2)

library(dplyr)

library(xlsx)

download.file(url="<https://community.watsonanalytics.com/wp-content/uploads/2015/03/WA_Fn-UseC_-HR-Employee-Attrition.xlsx>",

destfile="./data.xlsx",

mode="wb")

df <- read.xlsx("./data.xlsx",

sheetIndex=1,

header=TRUE,

colClasses=NA)

ggplot(filter(df, (YearsAtCompany >= 2) & (YearsAtCompany <= 5) & (JobLevel < 3)),

aes(x=factor(Department), y=MonthlyIncome, color=factor(Attrition))) +

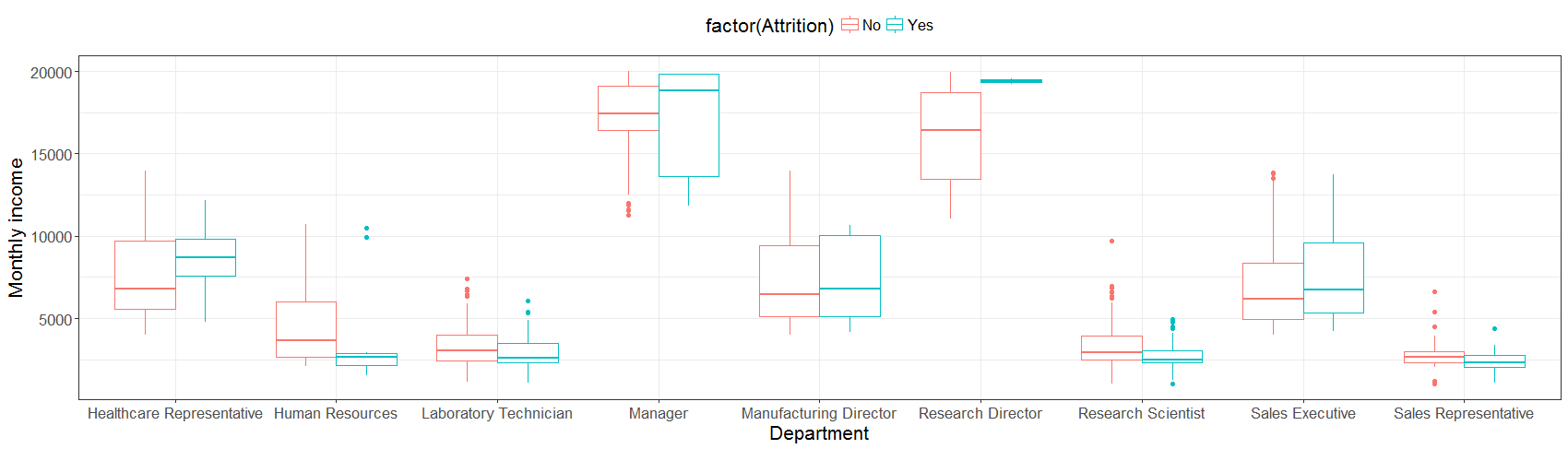
geom\_boxplot() +

xlab("Department") +

ylab("Monthly income") +

scale\_fill\_discrete(guide=guide\_legend(title="Attrition")) +

theme\_bw()



* R packages such as tm offer handy functions for dealing with text mining tasks. Sometimes text analysis on multiple languages are of great interest to employers have subsidiaries located in different countries. This can be done either via language-specific text analysis package (e.g., jiebaR for Chinese segmentation) or on-line translation. Following is a snippet on how to perform tokenization and term frequency count on English-Chinese text with jiebaR and tm.

library(jiebaR)

library(tm)

library(magrittr)

# sample text data.

df\_text <- data.frame(

text = c(

"工作好辛苦",

"每天都只有辛苦的工作",

"今天非常开心！",

"It is a great honor to work in Microsoft.",

"工作压力巨大。",

"吃了喜欢的拉面，开心！",

"weekend要工作，难过……",

"今天放假！",

"不想工作。"

),

mood = c( # P is positive and N is negative.

"N",

"N",

"P",

"N",

"P",

"P",

"N",

"P",

"N"

),

stringsAsFactors = FALSE

)

# firstly make it a corpus.

corp\_text <-

Corpus(VectorSource(df\_text$text)) %>%

tm\_map(removePunctuation) %>%

tm\_map(removeWords, stopwords("en"))

# instantiate a tokenier. Specify the Chinese stop words dictionary.

cutter2 <- worker(stop\_word = "<path\_to\_stop\_words\_file>", bylines = TRUE)

# Note as jiebaR does not provide methods for term frequency transformation from corpus, tm is used for doing this.

# customize a tokenizer function that can be embedded into tm functions.

# DocumentTermMatrix function.

jieba\_tokenizer <- function(d) {

unlist(segment(d[[1]], cutter2))

}

dtm\_text <-

DocumentTermMatrix(corp\_text,

control = list(wordLengths = c(1, Inf),

weighting = weightTf,

tokenize = jieba\_tokenizer))

# produce a data frame of document term frequency.

df\_dtf <-

as.data.frame(inspect(dtm\_text)) %>%

cbind(mood = df\_text$mood) %>%

arrange(mood) %T>%

{head(.) %>% print()}

<<DocumentTermMatrix (documents: 9, terms: 20)>>

Non-/sparse entries: 27/153

Sparsity : 85%

Maximal term length: 9

Weighting : term frequency (tf)

Terms

Docs great honor it microsoft weekend work ... <truncated>

1 0 0 0 0 0 0 ... <truncated>

2 0 0 0 0 0 0 ... <truncated>

3 0 0 0 0 0 0 ... <truncated>

4 1 1 1 1 0 1 ... <truncated>

5 0 0 0 0 0 0 ... <truncated>

6 0 0 0 0 0 0 ... <truncated>

7 0 0 0 0 1 0 ... <truncated>

8 0 0 0 0 0 0 ... <truncated>

9 0 0 0 0 0 0 ... <truncated>

Terms

Docs 好 工作 巨大 ... <truncated>

1 1 1 0 ... <truncated>

2 0 1 0 ... <truncated>

3 0 0 0 ... <truncated>

4 0 0 0 ... <truncated>

5 0 1 1 ... <truncated>

6 0 0 0 ... <truncated>

7 0 1 0 ... <truncated>

8 0 0 0 ... <truncated>

9 0 1 0 ... <truncated>

Terms

Docs 辛苦 难过

1 1 0

2 1 0

3 0 0

4 0 0

5 0 0

6 0 0

7 0 1

8 0 0

9 0 0

Complete version of the above analysis can be found on Github repository. Codes for the analytics are supplied and embedded in an R markdown document, which can be interactively executed step by step to aid replication and learning. Documents of various formats (e.g., PDF, html, Jupyter Notebook, etc.) can be produced directly from the markdown document.