Predicting Employee Attrition with Machine Learning

Reem Taye
Western Governors University
Master of Science in Data Analytics
February 2019

Contents

Abstract

Employee attrition is a costly issue facing employers in a market that is growing increasingly in favor of job seekers. This paper steps through the process of binary classification with regards to the issue of employee attrition. A dataset of simulated employee information is analyzed for the purposes of creating a predictive model to identify risk factors in possible employee turnover. SVM, Random Forests, Logistic Regression and Neural Networks are applied to the data and the null hypothesis is rejected in favor of significance between employees who leave organizations and their counterparts that remain. Several recommendations are made based on these findings including ideas to increase overall employee engagement to curb turnover.

Research Question:

Unemployment currently sits at 4% (Brainerd, 2019), which is lower than full employment (4.1 - 4.7%) in the United States. (Crook, 2015) A rate below full employment signals many of the issues facing employers today: a shrinking candidate pool, difficulty getting candidates to accept positions and losing skilled employees to better offers in the market. The costs associated with filling a vacancy rise the longer that position is held open, especially if the position was vacated suddenly by an employee's resignation. A low supply of candidates prompts counter-offers and increased costs associated with recruiting efforts, all of which contribute to the total cost of employee turnover. (Kelly, 2018)

As these expenses continue to rise, they put employers in a position of risk, specifically to their budgets and planning. (Insightlink, 2014) In order to cut the expense at the source, is it possible for employers to identify those employees more likely to resign? To answer this question, a null hypotheses and alternative hypothesis are proposed. The null hypothesis is that

there is no significant difference between employees who resign and employees who stay with an employer, and thus no actionable information can be retrieved that could reduce turnover costs or prevent loss of skilled employees. An alternative hypothesis is that there is a significant difference between employees who resign and employees who stay with an employer, which could possibly lead to actionable solutions that reduce recruitment and turnover costs by retaining more employees who would otherwise leave due to the analyzed criteria.

Data Collection:

Employee attrition is a difficult topic to analyze without access to employer data. Because such data contains personally identifying information, companies are not at liberty to openly share this information with the public. For this reason, the dataset used for this research is a simulated dataset provided by IBM for the purposes of analyzing employee attrition and retention within organizations. The sample dataset contained 35 variables including 34 independent variables encompassing employee demographics and engagement and 1 target variable (Attrition). The data was of a mixed nature with 1,470 total observations. IBM provides several datasets for analysis to the public via their website. Had the data not been readily available, this research may not have been possible due to the confidential nature of employee data. (IBM, 2015)

The advantage to using simulated data is that it's relatively clean, doesn't require intense transformations prior to analysis and contains no sensitive information that would bar wide-scale testing and analysis. In a study conducted by the National Science Foundation, simulated data (which is inherently based on real data), was found to model predictions with no statistical difference to real data. (Koperniak, 2017) The disadvantage here is that simulated data may not be actionable in all domains – employment changes by industry, level and scope. The IBM data

is related to attrition among employees in the organizational departments of Research & Development, Human Resources and Sales and my not be helpful in determining methodologies that could be applied outside of these departments.

Data Extraction & Preparation

Reviewing the Dataset

An initial review of the data via Microsoft Excel Power Query allowed for a quick examination of glaring issues and data characteristics. The data appeared to be of mixed type, with some ordinal features coded numerically. Reviewing the data in this way allows the analyst to plan for possible tests and manipulations needed by simply looking at the data. R provides the ability to view datasets with the head() or view() functions, however Power Query allows for quick manipulations to the data via an easy-to-use UI with no need for coding. Most business users will find Power Query easily accessible within the Data Tab of Microsoft Excel, so it's a tool with a low learning curve since the base understanding exists amongst most users.

For example, if the data required dummy coding to convert categorical variables to continuous or binary ones, Power Query can do it with a click of two buttons in the editor. Another click can change an entire column from 0/1 values to Yes/No based on the needs of the user. For the purposes of this research, text values for the variables Education Field and Department were edited to allow for simplicity in visualization. This level of quick manipulation can save a lot of time in gathering initial scope when preparing data for analysis, especially if a user is not fully experienced with R or another analytical tool. Aside from initial viewing of the data and transformations, Power Query lacks the robust capabilities of exploratory analysis that R contains. For this reason, the remaining data prep and analysis is completed in R. (Jain, 2017)

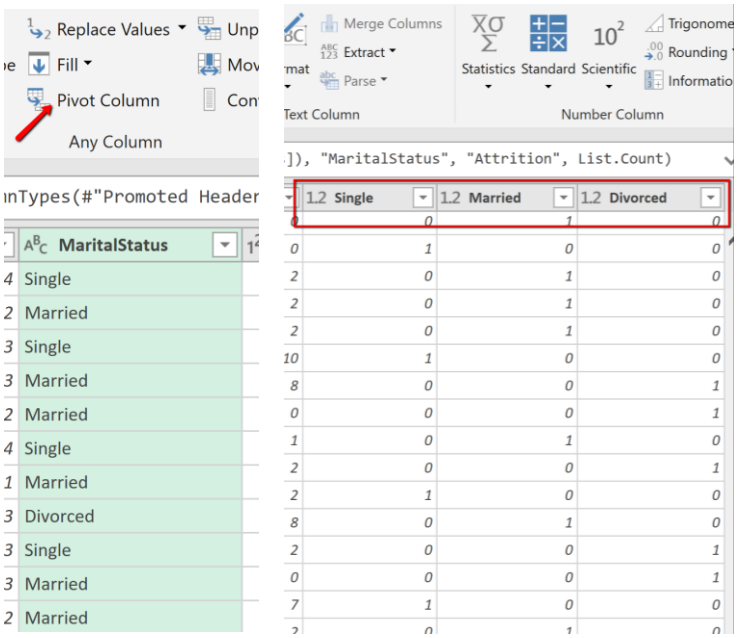

Figure 1 Power Query can create binary variables out of categorical variables with n-levels

Data Extraction

Data extraction consisted of manual download of an Excel file from the IBM website and converting an Excel document to CSV format for easy processing of the data. The file contained the dataset in addition to a second sheet with metadata related to the ordinal variables. CSV formatting removes extra Excel features that aren't needed for analysis and allows for the data to be viewed in multiple applications. While R can process Excel formats, this method was chosen to ensure only one dataset with no extra worksheets would be loaded into R for analysis. Rawd.csv was then loaded into R Studio using R version 3.5.2 and several packages that will be referenced throughout this research.

```
Load original dataset (no cleaning has been done)
```{r}
setwd("C:/[redacted]/WGU/C772")
raw <- read.csv("rawd.csv", header = TRUE)
```

Figure 2 Loading raw data with R Studio
```

Missing Values, Data Types and Constant Variables

Upon initial exploration with Microsoft Excel Power Query, the employee attrition dataset provided by IBM appeared to contain no missing values. To verify this, colsums() was used with is.na() in R to count the number of missing values per column.

```
24 colSums(is.na(rawd))
25
```

```
                    Age                Attrition           BusinessTravel                DailyRate               Department
                      0                        0                        0                        0                        0
       DistanceFromHome                Education          EducationField  EnvironmentSatisfaction                   Gender
                      0                        0                        0                        0                        0
             HourlyRate           JobInvolvement                 JobLevel                  JobRole            JobSatisfaction
                      0                        0                        0                        0                        0
          MaritalStatus            MonthlyIncome              MonthlyRate         NumCompaniesWorked                 OverTime
                      0                        0                        0                        0                        0
      PercentSalaryHike       PerformanceRating RelationshipSatisfaction       StockOptionLevel        TotalWorkingYears
                      0                        0                        0                        0                        0
  TrainingTimesLastYear          WorkLifeBalance           YearsAtCompany      YearsInCurrentRole  YearsSinceLastPromotion
                      0                        0                        0                        0                        0
   YearsWithCurrManager
                      0
```

Figure 3 Sum of missing values per column

The initial dataset contained information assigning numeric codes to several ordinal variables and nominal variables were left as is. A full listing of variables and their raw data types can be found in the Appendix. The ordinal variables were coded numerically except for BusinessTravel, which contained text values.

Variable	**Values**
Education	1-Below College/High School, 2-College, 3-Bachelors, 4-Masters, 5-Doctorate
EnvironmentSatisfaction JobInvolvement JobSatisfaction Relationship Satisfaction	1-Low, 2-Medium, 3-High, 4-Very High
PerformanceRating	1-Low, 2-Good, 3-Excellent, 4-Outstanding
WorkLifeBalance	1-Bad, 2-Good, 3-Better, 4-Best

Table 1 Ordinal Variables

R provides simple functions to examine raw data for cleaning and manipulation. The str() function returns a list of all columns in the dataset along with their respective data types as determined by R upon import. Over18 was listed as a factor with one level, which means it contains the same value for every row and thus isn't relevant for analysis. Running summary() on the data revealed that EmployeeCount and StandardHours also contained the same value for

every row. EmployeeNumber contained only unique values. All 4 of these variables were set to NULL, removing them from the data before further processing.

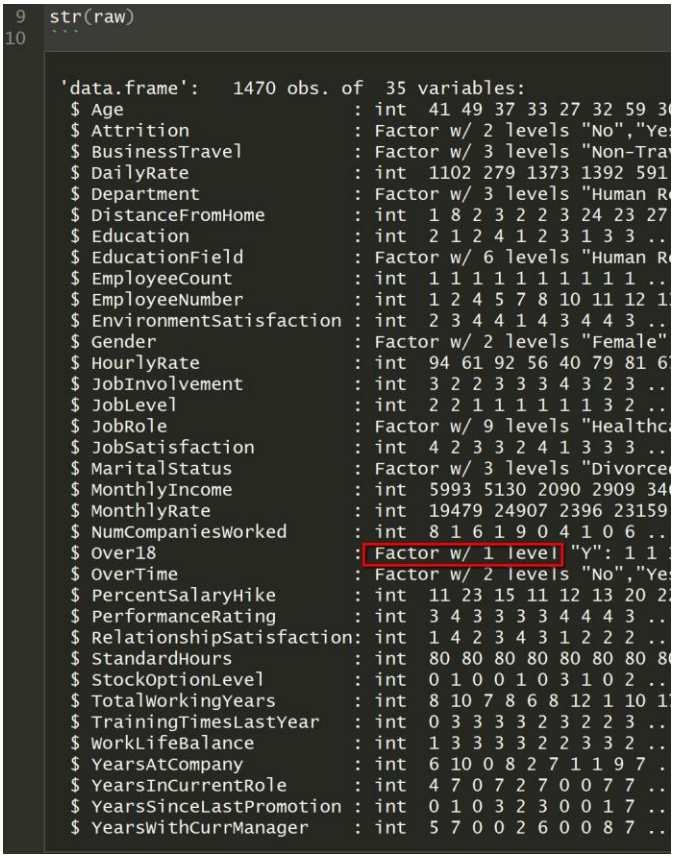

Figure 4 Str() function returns data types

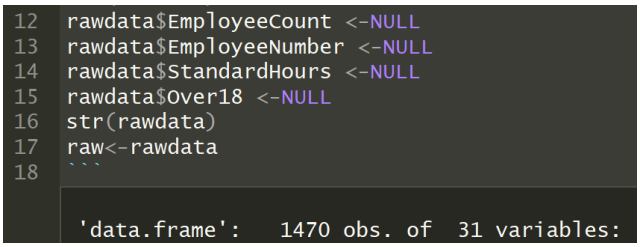

Figure 5 4 Variables removed, 31 remaining

Data Conversions

The mixed nature of the data meant that one of three approaches was required for further analysis. Continuous features could be converted to categorical by binning the data. Binning data may lead to a loss in information, however, so there is a tradeoff in standardizing the attributes in

this way. (Wainer, 2009) Categorical variables can be converted to binary 0/1 numeric variables by design coding: breaking out each attribute into several attributes (n-1 levels). Because each level must be broken out individually, this method can drastically increase dimensionality. The third option is to leave both types as they are and conduct a selection of mixed analysis techniques on the data. This way, no information is lost and an accurate exploration of the data can be completed.

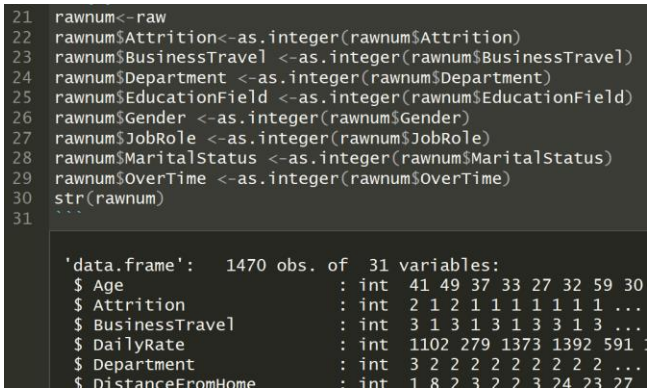

Figure 5 Numeric data conversions

Tools & Techniques

R boasts of a massive open source community and thousands of packages available for use. Its limitations lie within the processing power of the user's device, because it can do almost anything from an analytics perspective. The decision to use R for this research was a matter of preference and not an issue of functionality. Python, SAS and other tools could have handled all the testing and visualizations produced for this project.(Jain, 2017) The techniques chosen were selected based on the nature of the data and the requirements of the study. Tests for association and classification models are diverse in their scope and applications, and they continue to evolve. However, core tests like the Chi-Square test of Association and the Pearson correlation continue to be the standard methods for analyzing data of this nature. (Gonzalez-Chica, 2015)

Analysis

Univariate & Bivariate Exploration

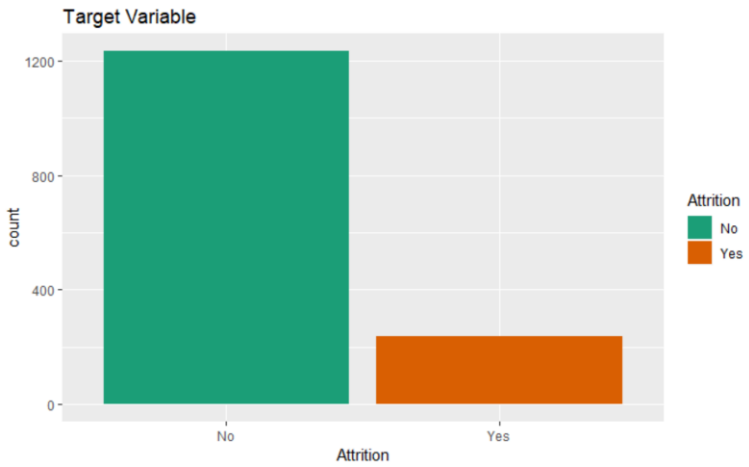

Figure 6 Target Variable: Attrition

The target variable is Attrition with 16% of the sample containing a positive event. While this isn't quite a rare event as it is over 10% of the events in the data (Chawla, n.d.), it was close enough to warrant exploration into oversampling methods.

The ggplot2 package was used to plot relationships of variables with Attrition. Variables were grouped by engagement, job features and demographics. In the engagement group, attrition is higher among those with a performance rating of three and significantly higher in lower income groups. Employees tend to leave more often if they've been with the company less than five years, but there doesn't appear to be any significance in the MonthlyRate, HourlyRate or JobInvolevement plots. In the demographics group, single employees have a much higher rate of attrition relative to married and divorced employees. In the job features group, employees who work overtime are significantly more likely to leave than their counterparts who do not work overtime. Further analysis is required prior to feature selection and modeling.

Figure 7 Variable distributions by the target

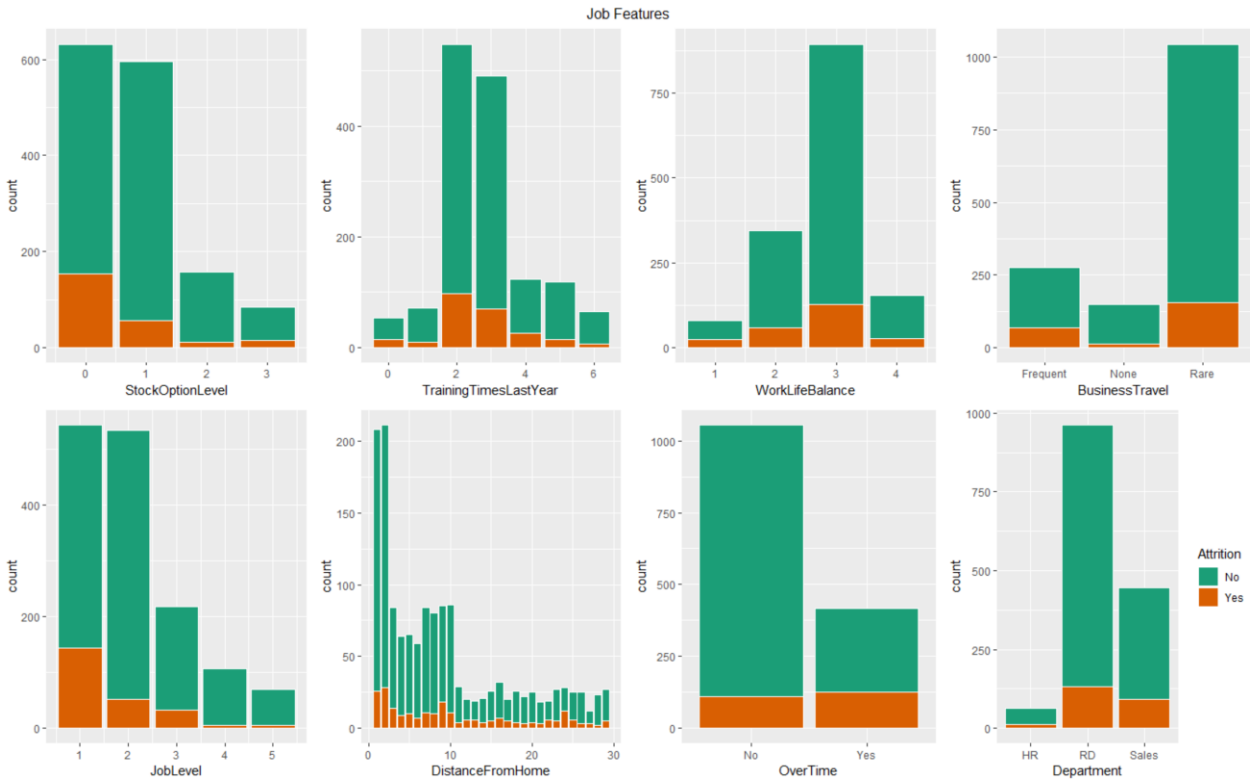

Figure 8 Plots continued

Reducing Dimensionality

Initial tests for correlation and association were done without changing data types for any of the predictors in order to minimize any losses of information that could result from binning or design coding. (Wainer, 2009) Before removing any attributes from the dataset, each attribute is evaluated against the target and against other predictors of its type. Chi-square tests for association were applied between the target and all categorical variables and again between categorical predictors. The null hypothesis for these tests is independence between variables. (Gonzalez-Chica, 2015) All p-values below a significance level of 0.05 are in black. The target variable shows some association with those variables and little to no association with gender, education, relationship satisfaction and performance rating. Job role and job level are both highly associated with three or more predictors. Associations between predictors were run by simply switching out the variable in the object att.

```
87  att<-raw$Attrition
88  l<-table(raw$BusinessTravel,att)
89  chisq.test(l) #X-squared = 24.182
90  cramersV(l)
```

Figure 9 Chi-Square and Cramers V Tests

```
        Pearson's Chi-squared test

data:  l
X-squared = 24.182, df = 2, p-value = 5.609e-06

[1] 0.12826
```

Figure 10 Test Results

Chi-Square Tests for Association with Attrition (α=0.05)				
Variable	**X^2**	**df**	**P-value**	**CramersV**
BusinessTravel	24.182	2	5.61E-06	0.12826
Department	10.796	2	0.004526	0.08569844
JobRole	86.19	8	2.80E-15	0.2421422
OverTime	87.564	1	2.20E-16	0.2440646
Gender	1.117	1	0.2906	0.02756522
MaritalStatus	46.164	2	9.46E-11	0.1772113
EducationField	16.025	5	0.006774	0.1044085
Education	3.074	4	0.5455	0.04572888
JobInvolvement	28.492	3	2.86E-06	0.1392204
JobLevel	72.529	4	6.64E-15	0.2221249
JobRole	86.19	8	2.80E-15	0.2421422
JobSatisfaction	17.505	3	0.0005563	0.1091248
EnvironmentSatisfaction	22.504	3	5.12E-05	0.1237286
RelationshipSatisfaction	5.2411	3	0.155	0.05971057
PerformanceRating	0.0001548	1	0.9901	0.0003244612
StockOptionLevel	60.598	3	4.38E-13	0.2030353
WorkLifeBalance	16.325	3	0.0009726	0.1053827

Table 2 Tests for association results

Chi-Square Tests for Association between Predictors (α=0.05)			
Variable 1	**Variable 2**	**P-Value**	**CramersV**
JobRole	Education	0.0380638	0.08989737
Department	EducationField	7.77E-214	0.590451
JobRole	EducationField	1.72E-155	0.3430071
JobRole	Gender	0.0419544	0.1044255
Department	JobLevel	1.97E-26	0.2185238
JobRole	JobLevel	0	0.5727782
EducationField	JobLevel	2.60E-07	0.1083664
JobRole	Department	0	0.9393926
JobRole	MaritalStatus	0.0424643	0.09567634
MaritalStatus	StockOptionLevel	2.30E-212	0.5826577

JobLevel	StockOptionLevel	0.0008844	0.08683264

Table 3 Redundancy tests between predictors

Based on the correlation values and Chi-Square p-values, several variables could be safely removed from the data model, thus reducing dimensionality. Visualizations of the predictors and Cramer's V statistics were computed. Results under 0.15 are very weak and are not considered to be dependencies for the sake of dimensionality reduction. (UToronto, n.d.) Results above 0.50 are redundant and signal that one of the variables should be removed from the model. From the categorical predictors tested, gender, education, relationship satisfaction and performance rating are be removed from the dataset due to no significant association with the target.

For the continuous predictors, a correlation plot using the Pearson method displays all linear correlations within the data. The decision was made to consider ordinal variables in both the continuous and categorical tests for maximum coverage of possible redundancy. A shapiro-wilk test confirms normality in the continuous data, verifying the choice to use the Pearson method over a rank-sum method like the Spearman coefficient. (NIST, n.d.) Strong correlations above 0.70 were found between the following variables:

- Total working years and monthly income
- Total working years and job level
- Years with current manager and years in current role
- Years with current manager and years at company
- Years at company and years in current role
- Job level and monthly income

```
Correlations
```{r}
##3: Create an initial correlation plot to explore relationships
library(dplyr)
library(corpcor)
library(corrplot)
attach(raw)
shapiro.test(YearsInCurrentRole)
shapiro.test(YearsSinceLastPromotion)
shapiro.test(YearsWithCurrManager)
shapiro.test(YearsAtCompany)
shapiro.test(TotalWorkingYears)
shapiro.test(Age)
shapiro.test(DailyRate)
shapiro.test(MonthlyRate)
shapiro.test(MonthlyIncome)
shapiro.test(NumCompaniesWorked)
shapiro.test(DistanceFromHome)
raw.numeric <- select_if(raw, is.numeric)
#raw.numeric<-subset(raw.numeric, select=-c(4,5,7,8,9,14:16,19))
colnames(raw.numeric)
correlations <-cor(raw.numeric)
corrplot(correlations, tl.srt=25,type="lower", method="number",
         tl.col="black", sig.level=0.05, insig="blank",tl.cex=.8,
         main="Numeric & Ordinal Correlations",mar=c(0,0,2,0))
#as.data.frame(correlations)
```
```

Figure 11 Normalcy testing and correlation plots

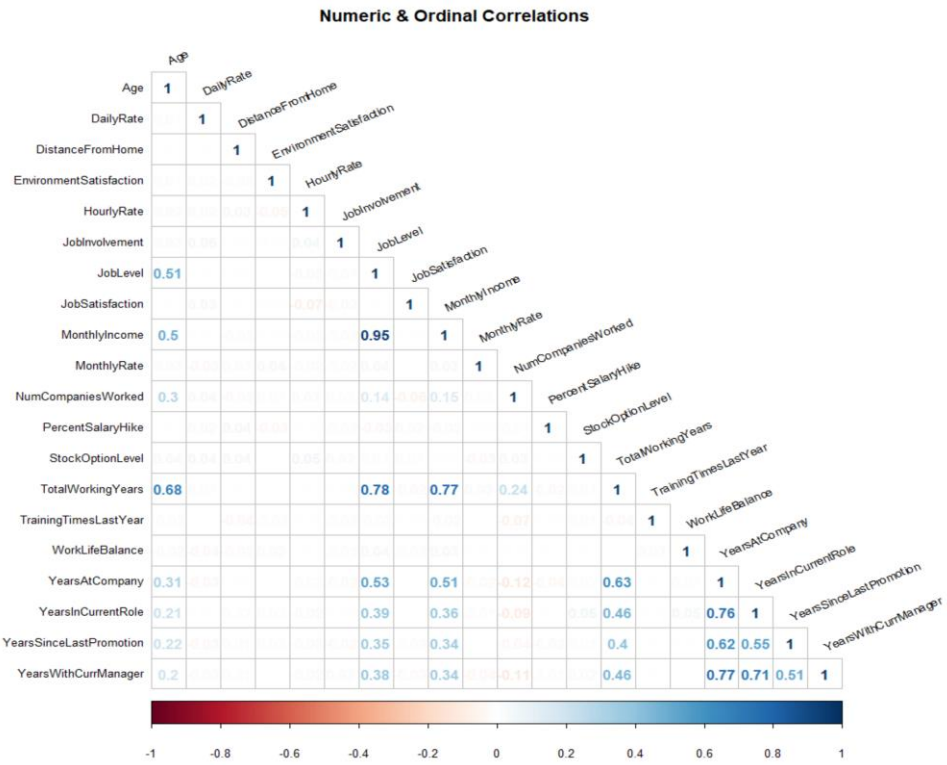

Figure 12 Pearson Correlation Matrix

The findcorrelation function in R scans through all correlations of numeric data type and returns those with a correlation above the designated cutoff. A cutoff correlation of 0.70 was selected to flag all numeric columns for removal. Years at company, job level, total working years and years with current manager were flagged.

```
223 Continuous Vars
224 ```{r}
225 ##5: Explore relationships between highly correlated variables
226 library(caret)
227 set.seed(11)
228 Flagged <- findCorrelation(correlations,0.7, verbose=FALSE, names=TRUE,exact=TRUE) #flag
229 print(Flagged) #Flagged Columns:"TotalWorkingYears","JobLevel","YearsAtCompany","YearsInC
230 ```

 [1] "TotalWorkingYears"  "YearsAtCompany"     "JobLevel"           "YearsInCurrentRole"
```

Figure 13 findcorrelation() tests

To measure association with the target, contingency tables Chi-Square were used. Results for monthly income and monthly rate were extremely high compared to the favorable association found in correlated variable job level. Total working years, years with current manager, years at company and years in current role exhibited similar associations with the target.

Of the correlated continuous predictors, daily rate and monthly rate are irrelevant and will be removed from the dataset. Monthly income is highly correlated with job level but exhibits unfavorable association with the target, so it's deemed redundant and will be removed. Of the three correlated years variables, years at company shows the highest association with the target and will be retained. Years in current role and years with current manager will be removed due to multicollinearity.

Chi-Square Tests for Association with Attrition (α=0.05)		
Variable	**P-Value**	**CramersV**
Age	2.58E-09	0.2847304
DailyRate	0.62028	0.7699804
DistanceFromHome	0.09525314	0.1611365
HourlyRate	0.4105697	0.2213575
MonthlyIncome	0.7085811	0.9472567
MonthlyRate	0.651263	0.9775298
NumCompaniesWorked	0.002249178	0.1323374
PercentSalaryHike	0.4989552	0.09530741
TotalWorkingYears	1.59E-10	0.288442
TrainingTimesLastYear	0.01914773	0.1015072
YearsAtCompany	2.84E-07	0.2547358
YearsInCurrentRole	4.06E-07	0.2091462
YearsSinceLastPromotion	0.1119339	0.1219037
YearsWithCurrentManager	3.41E-09	0.2252998

Table 4 Continuous variables tests for association

With an initial dataset containing 35 variables, addressing the issue of high dimensionality is crucial in order to prevent overfitting of the model. In this process, the goal was to reduce the number of predictor variables by removing those deemed to be irrelevant or redundant. Filtering methods utilized in this research reduced dimensionality by eliminating 13 irrelevant and/or redundant predictors.

Feature Selection

Various algorithms can be useful in determining feature importance for improved model performance. Automated feature selection techniques are advantageous for their ability to generalize well and produce accurate results in less time than manual selection. Unfortunately, there is no "one size fits all" in the realm of automated feature selection, and one model may not necessarily yield an outcome that could be applied to another model. A random forest algorithm was used to determine most important features prior to modeling. Random forest combines multiple decision trees in a series of nodes based on purity, which is a measure of variance. The purpose of feature selection is to determine the combination of features that yield the most accurate predictions in the models. For this task, the Caret package in R was utlizied to find the top features that contribute to the model.

```
set.seed(11)
importance.model.glm <- train(Attrition~., data=p, method="glm", trControl=raw.ctrl)
set.seed(11)
importance.model.svm <- train(Attrition~., data=p, method="svmRadial", trControl=raw.ctrl)
```
Feature selection part 2
```{r}
# var importance
redundant.glm <- varImp(importance.model.glm, scale=TRUE)
#redundant.svm <- varImp(importance.model.svm, scale=TRUE)
redundant.rf <- varImp(importance.model.rf, scale=TRUE)
#redundant.nn <- varImp(importance.model.nn, scale=TRUE)
#redundant.pca <-varImp(importance.model.pca, scale=TRUE)
c<-ggplot(redundant.glm,mapping=aes(redundant.glm))+ggtitle("GLM Feature Importance")
#d<-ggplot(redundant.svm,mapping=aes(redundant.svm))+ggtitle("SVM Feature Importance")
e<-ggplot(redundant.rf,mapping=aes(redundant.rf))+ggtitle("RF Feature Importance")#+geom_ba
#e<-ggplot(rf3,aes(rf3))+ggtitle("RF Feature Importance")+geom_bar()+scale_color_brewer(pal
#f<-ggplot(redundant.nn,mapping=aes(redundant.nn))+ggtitle("NN Feature Importance")
#grid.arrange(c,d,ncol=2,top="Most important Features: GLM & SVM")
grid.arrange(e,c,ncol=2,top="Most important Features: RF & GLM")
```

Figure 14 Feature selection code

Importance was scaled using the scale=TRUE parameter to standardize the comparison between features. The random forest splits features down into smaller branches, with the least important features being levels within features. Age, total working years, hourly rate, distance from home, overtime, years at company and percent salary hike were the most important features based on this method, however the remaining features all contributed a significant amount to the model. For this reason, two datasets will be created: 1 with all remaining features after dimension reduction and one with the top 15 variables from the RF feature importance. Both datasets will be modeled and measured for accuracy against the test data.

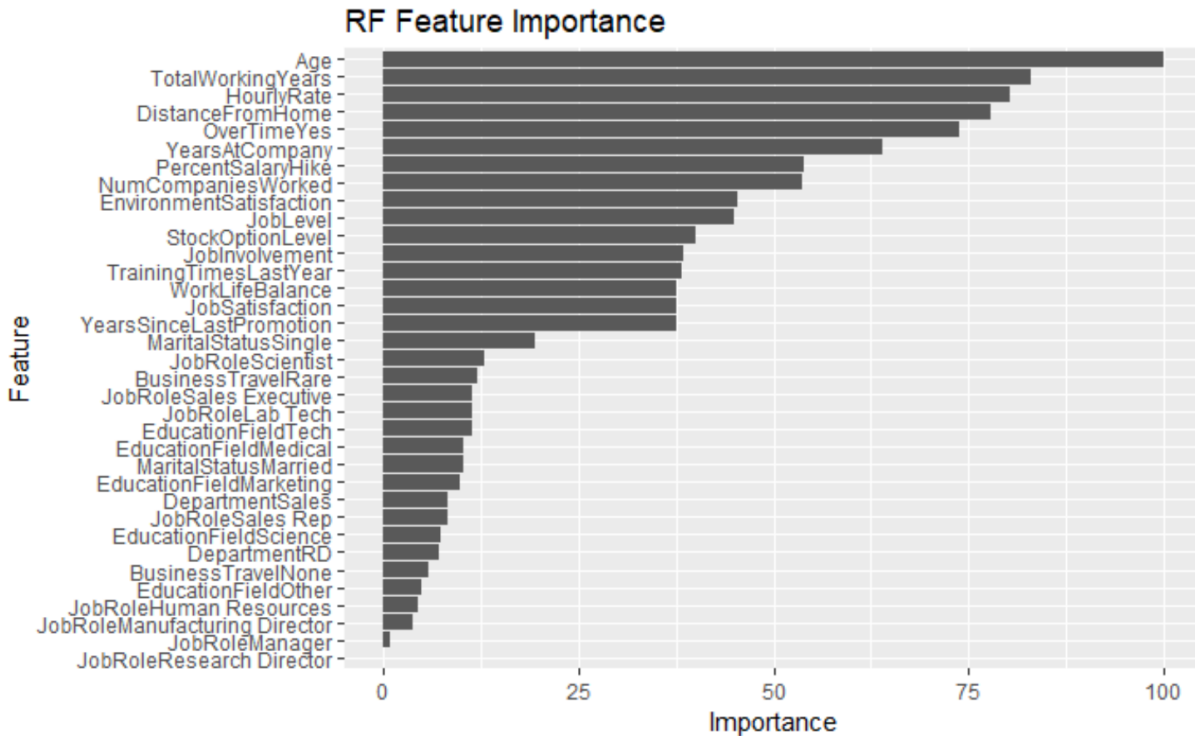

Figure 15 Feature Importance Rankings

```
> colnames(raw)#dataset1
 [1] "Age"                      "Attrition"
 [3] "BusinessTravel"           "Department"
 [5] "DistanceFromHome"         "EducationField"
 [7] "EnvironmentSatisfaction" "HourlyRate"
 [9] "JobInvolvement"           "JobLevel"
[11] "JobRole"                  "JobSatisfaction"
[13] "MaritalStatus"            "NumCompaniesWorked"
[15] "OverTime"                 "PercentSalaryHike"
[17] "StockOptionLevel"         "TotalWorkingYears"
[19] "TrainingTimesLastYear"    "WorkLifeBalance"
[21] "YearsAtCompany"           "YearsSinceLastPromotion"
> colnames(raw.5)#RF feature dataset
 [1] "Age"                      "Attrition"
 [3] "TotalWorkingYears"        "HourlyRate"
 [5] "DistanceFromHome"         "OverTime"
 [7] "YearsAtCompany"           "PercentSalaryHike"
 [9] "NumCompaniesWorked"       "EnvironmentSatisfaction"
[11] "JobLevel"                 "StockOptionLevel"
[13] "JobInvolvement"           "TrainingTimesLastYear"
[15] "WorkLifeBalance"          "JobSatisfaction"
>
```

Figure 16 Final datasets to be used in the model

Predictive Models: SVM, Logistic Regression, Random Forest, Neural Network

To account for the imbalance in the data, a smooth weight of evidence method was used to oversample and under-sample the positive and negative events respectively. Oversampling is necessary to rebalance a dataset when one event appears disproportionately to the other in a dataset. While the 16% positive event (attrition) is higher than what would be considered rare, balancing the data will still help prevent overfitting towards the negative outcome. (Chawla, n.d.)

The data was split into a 75:25 training to test ratio. SMOTE was applied only to the training data and the models were validated with the test data. Using this method created a training data set with an even split between observations *with* the event and observations *without* the event. The test dataset remains unchanged to ensure validation of the prediction results. (Chawla, n.d.)

```
381 library(e1071)
382 library(DMwR)
383 library(rpart)
384 library(ROSE)
385 #Oversampling Trials
386 #SPLIT
387 split0<-createDataPartition(raw$Attrition,p = .75, list = FALSE) #raw
388 traw <- raw[split0, ] #training
389 sraw  <- raw[-split0, ] #test
390 #split4 <- createDataPartition(raw.4$Attrition,p = .75, list = FALSE)
391 #traw4 <- raw.4[split4, ] #training
392 #sraw4  <- raw.4[-split4, ] #test
393 set.seed(11)
394 split5 <- createDataPartition(raw.5$Attrition,p = .75, list = FALSE)
395 traw5 <- raw.5[split5, ] #training
396 sraw5  <- raw.5[-split5, ] #test
397 set.seed(11)
398 #split6 <- createDataPartition(raw.6$Attrition ,p = .75, list = FALSE
399 #traw6 <- raw.6[split6, ] #training
400 #sraw6  <- raw.6[-split6, ] #test
401 table(raw.5$Attrition)
402 table(traw5$Attrition)
403
404 #SMOTE
405 library(DMwR)
406 os0 <-SMOTE(Attrition ~ ., traw, perc.over = 100, perc.under=200)
407 #os4 <- SMOTE(Attrition ~ ., traw4, perc.over = 100, perc.under=200)
408 os5 <- SMOTE(Attrition ~ ., traw5, perc.over = 100, perc.under=200)
409 #os6 <- SMOTE(Attrition ~ ., traw6, perc.over = 100, perc.under=200)
410
411 prop.table(table(os0$Attrition))
412 prop.table(table(os5$Attrition))
```

Figure 17 Data splitting and SMOTE

```
411 prop.table(table(os0$Attrition))
412 prop.table(table(os5$Attrition))
413 #prop.table(table(os6$Attrition))
414 ```


   No  Yes
 1233  237

  No Yes
 925 178

  No Yes
 0.5 0.5

  No Yes
 0.5 0.5
```

Figure 18 SMOTE results

Cross validation within the model training was eliminated because of SMOTE. Cross validation on replicated or oversampled data may not yield accurate results since an observation could appear in both the train and test sets. In a modeling situation where oversampling isn't needed, K-fold cross validation could help train models with relatively low observations. (Chawla, n.d.)

Five machine learning algorithms were selected for classification: Support Vector Machines, Logistic Regression, Neural Networks, Random Forest and Decision Tree. All models were applied using the Caret Package train function in R, but there are many other ways to train models. One of the biggest advantages of R is the constantly improving packages that allow any type of analysis depending on the nature of the data. A short description of each technique follows:

Support Vector Machines: SVM is a class separation technique that attempts to divide observations within an n-dimensional hyperplane through transformations called kernels. SVM models can be tuned in R with the cost and gamma parameters, though tuning this model can require extra processing time. A higher cost leads to a more complex split of the data and less misclassification errors. Gamma refers to the distance an observation in the plot needs to be for consideration in the calculation. Low gamma parameters will consider a larger spread of data. (Patel, 2017)

Logistic Regression: This method is a classic standard for binary classifications with any type of predictor data where the linear relationship is between the logits and the predictors. Even with the emergence of new and complex algorithms available for binary classification, logistic regression continues to be an excellent method for classification. (Penn State, 2019)

Neural Networks: These algorithms are colloquially referred to as modeling brain activity for classification and regression and can classify much of what the senses can feel images, text, sound and more. Because neural networks are designed for highly complex classifications, they’re suitable for a wide array of study and are included as a mechanism for comparison purposes in this study. (Nielsen, 2015)

Random Forest and Decision Tree: Decision trees model a conditional decision-making process, breaking down features into smaller groups and modeling the outcomes that cascade from those branches. A random forest consists of hundreds of decision trees combined at random, which provides immense diversity in variation and a better model fit overall. Random forest models are great for avoiding overfitting for this reason. In addition to being an excellent model for classification, this method can also be used to measure feature importance, as displayed earlier in this report. (Breiman, 2002)

Once the models completed training, predictions were applied to the test set for each model. Results of the predictions show a significantly better model output on the RF Features dataset, indicating that the feature selection improved the model versus the All Features dataset. Logistic Regression and Support Vector Machines had the best performance among the models in the RF Features data. Interestingly, the Random Forest model performed better under on the All Features dataset.

Model Performance				
	All Features Data		RF Features Data	
Model	**AUC**	**Accuracy**	**AUC**	**Accuracy**
GLM	0.751	0.7193	0.766	0.7684
SVM	0.747	0.7357	0.767	0.8038
NN	0.740	0.7248	0.708	0.7057
RF	0.748	0.8065	0.73	0.8338
DT	0.575	0.7466	0.603	0.782

Table 5 Predictive model performance

Performance was measured based on the area under the curve (AUC) of a ROC curve. For a binary classifier and imbalanced events, AUC is a much more accurate measure of performance than accuracy because it measures specificity and sensitivity. AUC doesn't require adjustment due to oversampling since it doesn't consider the intercept in its calculations, which makes it an easy and quick test for performance as well. Accuracy is not a suitable measure for this dataset due to the probability of a positive event being very low in comparison to the

negative event. Selecting a model based on accuracy may disregard sensitivity and specificity, yielding too many false positives and false negatives in the model.

Both datasets, the RF feature selection set and the original raw dataset were plotted on a ROC curve.

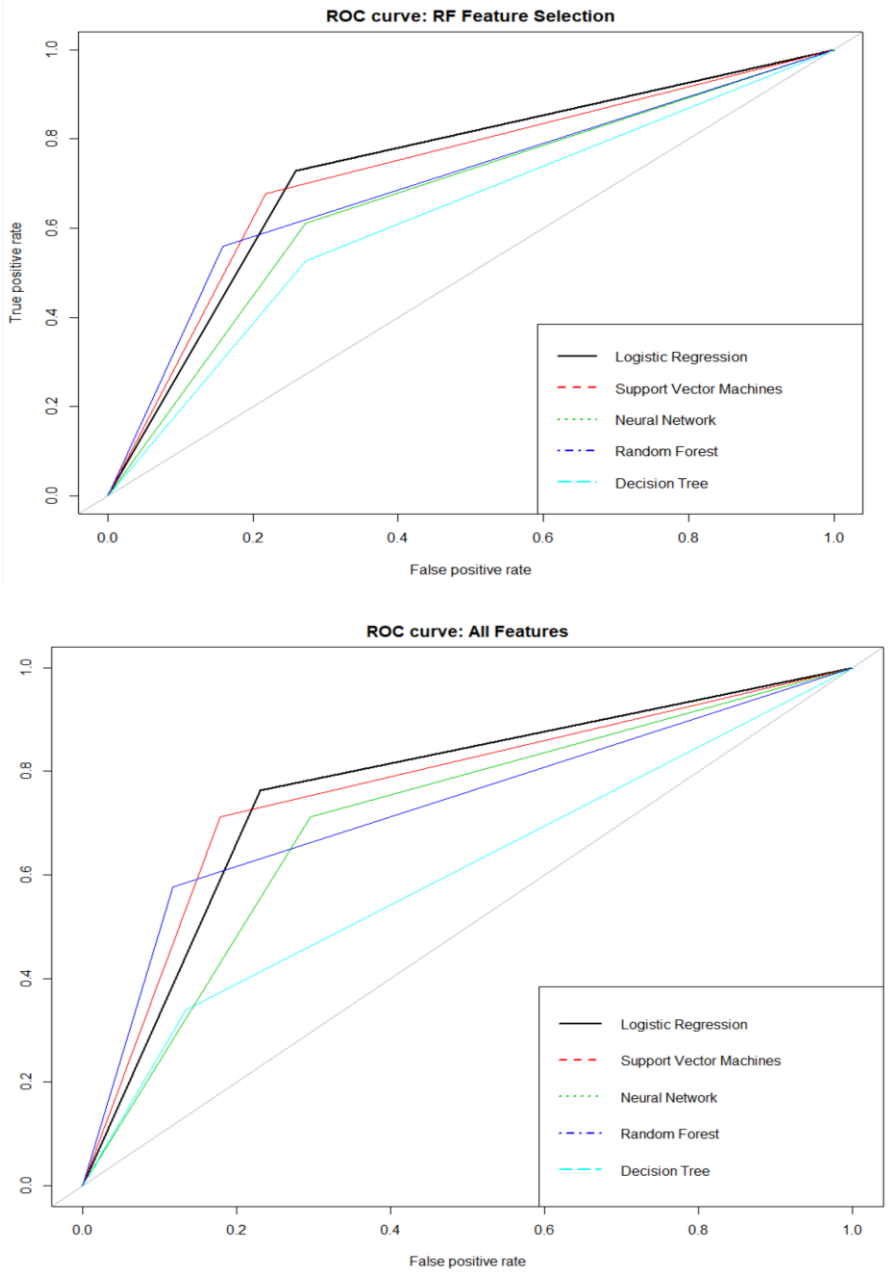

Figure 19 ROC curve results for both datasets

Data Summary and Implications

The null hypothesis at the start of this research was that there is no significant difference between employees who resign and employees who stay with an employer. Based on the analysis conducted, this research rejects the null hypothesis in favor of the alternative hypothesis: there is a significant difference between employees who leave and employees who stay. The process of feature selection identified a combination of 15 risk factors that may contribute to increased employee attrition. Employees are more likely to leave when they're still relatively new to an organization, so employers could take steps to ensure onboarding and new-hire programs are improved to reduce the learning curve that comes with starting a new job. A significantly higher number of low-level employees resign vs. everyone else, so a system seeking to promote from within could inspire low-level employees to go for a promotion instead of moving diagonally between organizations, which has become a normal practice in the market today. (Kelly, 2018) These risk factors could be further broken into clusters from which action plans are developed to craft counter-offers for employees that go past standard financial compensation.

A model like this could be implemented within a company's HR and recruiting teams to help flag those employees with the risk factor markers as indicated in the research, and a customized actionable plan could be developed based on each employee and their risk factors. Furthermore, this practice could be automated using tools like R and Python to allow for repeatability and consistency in reporting. Some companies are already offering these types of analytics as part of other suites of services such as timekeeping software and payroll/benefits services. Alice is a recent startup that aims to give hourly employees access to pre-tax benefits programs normally reserved for salaried employees, which they claim now leads to a 30% reduction in employee turnover. (Rosenberg, 2019)

Further Research

Recommendations based on risk factors may provide effective delay in employee attritions, however further research into actionable recommendations is needed. Employers would need to document all instances of retention activity and whether the employee remained after some time. This information could not only improve the existing model but could lead to improvements in overall employee retention that cannot be measured by this model. (For example: an employee resigning over pay may still leave after getting a raise). Evaluation on action taken and how long employees are retained for would open this up for much more interesting research.

Additionally, while this study is titled “Predicting Employee Attrition with Machine Learning”, it only applies to the departments supplied in the data (R&D, Sales and HR). A study conducted on a dataset (real or simulated) encompassing an entire corporation could allow for insights that have been deemed out of the scope of this study, such as attrition rate differences between hourly and salaried workers, and the impact of a work from home program on overall satisfaction and retention. A larger study could reduce costs of turnover for the employer and even possibly aid in developing programs to increase engagement and satisfaction across the board for all employees.

References

1. Anderson-Darling and Shapiro-Wilk tests. (n.d.). Retrieved February 24, 2019, from https://www.itl.nist.gov/div898/handbook/prc/section2/prc213.htm
2. Brainerd, J. (2019, February). Retrieved February 24, 2019, from http://www.ncsl.org/research/labor-and-employment/national-employment-monthly-update.aspx
3. Breiman, L. (2002). Random Forests Leo Breiman and Adele Cutler. Retrieved February 24, 2019, from https://www.stat.berkeley.edu/~breiman/RandomForests/cc_papers.htm
4. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (n.d.). SMOTE: Synthetic Minority Over-sampling Technique. Retrieved February 24, 2019, from https://www.jair.org/index.php/jair/article/view/10302
5. Crook, C. (2015, April 10). Full Employment. Retrieved February 24, 2019, from https://www.bloomberg.com/quicktake/full-employment
6. Crosstabulation with Nominal Variables. (n.d.). Retrieved February 24, 2019, from http://groups.chass.utoronto.ca/pol242/Labs/LM-3A/LM-3A_content.htm
7. Further Topics on Logistic Regression. (n.d.). Retrieved February 24, 2019, from https://newonlinecourses.science.psu.edu/stat504/node/217/
8. Gonzalez-Chica, D. A., Bastos, J. L., Duquia, R. P., Bonamigo, R. R., & Martínez-Mesa, J. (2015). Test of association: which one is the most appropriate for my study?. *Anais brasileiros de dermatologia*, *90*(4), 523-8.
9. HR Employee Attrition and Performance. (2015, April 11). Retrieved February 14, 2019, from https://www.ibm.com/communities/analytics/watson-analytics-blog/hr-employee-attrition/

10. Insightlink: What Does Employee Turnover Cost You? (2014, November 12). Retrieved February 24, 2019, from https://www.insightlink.com/blog/calculating-the-cost-of-employee-turnover.cfm
11. Jain, K., Shaikh, F., Dar, P., & Srivastava, T. (2017, September 12). Python vs. R (vs. SAS) - which tool should I learn? https://www.analyticsvidhya.com/blog/2017/09/sas-vs-vs-python-tool-learn/
12. Kelly, J. (2018, June 15). We Are Nearing A New Paradigm In The Job Market Where Employees Will Have The Power. Retrieved February 24, 2019, from https://www.forbes.com/sites/jackkelly/2018/06/15/we-are-nearing-a-new-paradigm-in-the-job-market-where-employees-will-have-the-power/#5a4cdb8839f4
13. Koperniak, S., & Institute for Data. (2017, March 03). Artificial data give the same results as real data - without compromising privacy. Retrieved February 10, 2019, from http://news.mit.edu/2017/artificial-data-give-same-results-as-real-data-0303
14. Nielsen, Michael A. "Neural Networks and Deep Learning", Determination Press, 2015 http://static.latexstudio.net/article/2018/0912/neuralnetworksanddeeplearning.pdf
15. Pochet, N. L., & Suykens, J. A. (2006, May 19). Support vector machines versus logistic regression: Improving prospective performance in clinical decision-making. Retrieved February 9, 2019, from https://obgyn.onlinelibrary.wiley.com/doi/full/10.1002/uog.2791
16. Rosenberg, T. (2019, February 12). Making an App to Make a Difference. Retrieved February 24, 2019, from https://www.nytimes.com/2019/02/12/opinion/start-ups-apps-social-impact.html
17. Wainer, Gessaroli, and Verdi (2009). Finding What Is Not There through the Unfortunate Binning of Results: The Mendel Effect, *Chance Magazine*, Vol 19, No.1, pp. 49 -52.

Appendix A: Data Types

Variable	Raw Data Type
Age	Continuous
Attrition	Binary
BusinessTravel	Ordinal w/3 levels
DailyRate	Continuous
Department	Nominal
DistanceFromHome	Continuous
Education	Ordinal
EducationField	Continuous
EnvironmentSatisfaction	Ordinal Coded Numeric
Gender	Binary
HourlyRate	Continuous
JobInvolvement	Ordinal Coded Numeric
JobLevel	Ordinal Coded Numeric
JobRole	Nominal
JobSatisfaction	Ordinal Coded Numeric
MaritalStatus	Nominal
MonthlyIncome	Continuous
MonthlyRate	Continuous
NumCompaniesWorked	Continuous
OverTime	Binary
PercentSalaryHike	Continuous
PerformanceRating	Ordinal Coded Numeric
RelationshipSatisfaction	Ordinal Coded Numeric
StockOptionLevel	Ordinal Coded Numeric
TotalWorkingYears	Continuous
TrainingTimesLastYear	Continuous
WorkLifeBalance	Ordinal Coded Numeric
YearsAtCompany	Continuous
YearsInCurrentRole	Continuous
YearsSinceLastPromotion	Continuous
YearsWithCurrManager	Continuous

Appendix B: Code

```
##1: Load
setwd("C://Users/reem/Dropbox/WGU/C772")
rawdata  <- read.csv("wa.csv", header = TRUE)
names(rawdata)
summary(rawdata)
#str(rawdata)
rawdata$EmployeeCount <-NULL
rawdata$EmployeeNumber <-NULL
rawdata$StandardHours <-NULL
rawdata$Over18 <-NULL
raw<-rawdata
##2: Univariate & Bivariate Exploration
library(RColorBrewer)
library(ggplot2)
library(gridExtra)
cc<- scale_fill_brewer(palette="Dark2")
ggplot(raw,aes(Attrition,fill=Attrition))+geom_bar()+cc+ggtitle("Target Variable")
ages<- ggplot(raw,aes(Age,fill=Attrition))+geom_bar()+guides(fill=FALSE)+cc
commute <- ggplot(raw,aes(DistanceFromHome,fill=Attrition))+geom_bar()+guides(fill=FALSE)+cc
dept <- ggplot(raw,aes(Department,fill = Attrition))+geom_bar()+cc
Edu <- ggplot(raw,aes(Education,fill=Attrition))+geom_bar()+guides(fill=FALSE)+cc#+scale_x_discr
EduF <- ggplot(raw,aes(EducationField,fill=Attrition))+geom_bar()+guides(fill=FALSE)+cc
Env <- ggplot(raw,aes(raw$EnvironmentSatisfaction,fill=Attrition))+geom_bar()+guides(fill=FALSE)
HRate <- ggplot(raw,aes(HourlyRate,fill=Attrition))+geom_bar()+cc+guides(fill=FALSE)
Income <- ggplot(raw,aes(MonthlyIncome,fill=Attrition))+geom_histogram()+cc+guides(fill=FALSE)
JInv <- ggplot(raw,aes(JobInvolvement,fill=Attrition))+geom_bar()+guides(fill=FALSE)+cc#+scale_x
JLev <- ggplot(raw,aes(JobLevel,fill=Attrition))+geom_bar()+cc+guides(fill=FALSE)
JSat <- ggplot(raw,aes(JobSatisfaction,fill=Attrition))+geom_bar()+guides(fill=FALSE)+cc#+scale_
MF <- ggplot(raw,aes(Gender,fill=Attrition))+geom_bar()+guides(fill=FALSE)+cc
MRate <- ggplot(raw,aes(MonthlyRate,fill=Attrition))+geom_histogram()+cc+guides(fill=FALSE)
NumC <- ggplot(raw,aes(NumCompaniesWorked,fill=Attrition))+geom_histogram()+cc
OT <- ggplot(raw,aes(OverTime,fill=Attrition))+geom_bar()+cc+guides(fill=FALSE)
Option <- ggplot(raw,aes(StockOptionLevel,fill = Attrition))+geom_bar()+cc+guides(fill=FALSE)
RSat <- ggplot(raw,aes(RelationshipSatisfaction,fill = Attrition))+geom_bar()+guides(fill=FALSE)
Raise <- ggplot(raw,aes(PercentSalaryHike,Attrition))+geom_point(size=4,alpha = 0.01)+cc
Score <- ggplot(raw,aes(PerformanceRating,fill = Attrition))+geom_bar()+guides(fill=FALSE)+cc#+s
Status <- ggplot(raw,aes(MaritalStatus,fill=Attrition))+geom_bar()+cc+guides(fill=FALSE)
trav <- ggplot(raw,aes(BusinessTravel,fill=Attrition))+geom_bar()+guides(fill=FALSE)+cc
```

```r
Trning <- ggplot(raw,aes(TrainingTimesLastYear,fill = Attrition))+geom_bar()+cc+guides(fill=FALSE)
Wlb <- ggplot(raw,aes(WorkLifeBalance,fill = Attrition))+geom_bar()+cc+guides(fill=FALSE)#+scale_x_discrete(limit = c("1", "2"
YrAtCom <- ggplot(raw,aes(YearsAtCompany,fill = Attrition))+geom_bar()+cc
YrInCurr <- ggplot(raw,aes(YearsInCurrentRole,fill = Attrition))+geom_bar()+guides(fill=FALSE)+cc
YrsSinceProm <- ggplot(raw,aes(YearsSinceLastPromotion,fill = Attrition))+geom_bar()+guides(fill=FALSE)+cc
YrsCurrMan <- ggplot(raw,aes(YearsWithCurrManager,fill = Attrition))+geom_bar()+guides(fill=FALSE)+cc
grid.arrange(Status,ages,Edu,EduF,MF, NumC,ncol=3,top = "Demographics")
grid.arrange(Option,Trning,Wlb,trav,JLev, commute,OT,dept,ncol=4,top = "Job Features")
grid.arrange(Score,JInv,JSat,RSat,Income,MRate,HRate,Env,YrInCurr,YrsSinceProm,YrsCurrMan,YrAtCom,ncol=3,top = "Engagement")
#Outliers
out <- boxplot.stats(raw$TotalWorkingYears)$out  # outlier values.
boxplot(raw$TotalWorkingYears, main="TotalWorkingYrs", boxwex=0.1)
mtext(paste("Outliers: ", paste(out, collapse=", ")), cex=0.6)
colnames(raw.numeric)
zz <- lm(Age ~ ., data=raw.numeric)
cooksd <- cooks.distance(zz)
plot(cooksd, pch="*", cex=2, main="Influential Obs by Cooks distance")  # plot cook's distance
abline(h = 5*mean(cooksd, na.rm=T), col="red")  # add cutoff line
text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>5*mean(cooksd, na.rm=T),names(cooksd),""), col="red")  # add labels
#Association Tests w/Target
##4: Association tests w/Target
library(lsr)
att<-raw$Attrition
l<-table(raw$BusinessTravel,att)
chisq.test(l) #X-squared = 24.182, df = 2, p-value = 5.609e-06
cramersV(l)
m<-table(raw$Department,att)
n<-table(raw$JobRole,att)
o<-table(raw$OverTime,att)
p<-table(raw$Gender,att)
q<-table(raw$MaritalStatus,att)
r<-table(raw$EducationField,att)
s<-table(raw$Education,att)
t<-table(raw$JobInvolvement,att)
u<-table(raw$JobLevel,att)
v<-table(raw$JobRole,att)
w<-table(raw$JobSatisfaction,att)
x<-table(raw$EnvironmentSatisfaction,att)
```

```
x<-table(raw$EnvironmentSatisfaction,att)
y<-table(raw$RelationshipSatisfaction,att)
z<-table(raw$PerformanceRating,att)
aa<-table(raw$StockOptionLevel,att)
bb<-table(raw$WorkLifeBalance,att)
chisq.test(m) #X-squared = 10.796, df = 2,
chisq.test(n) #X-squared = 86.19, df = 8, p
chisq.test(o) #X-squared = 87.564, df = 1,
chisq.test(p) #X-squared = 1.117, df = 1, p
chisq.test(q) #X-squared = 46.164, df = 2,
chisq.test(r) #X-squared = 16.025, df = 5,
chisq.test(s)#X-squared = 3.074, df = 4, p-
chisq.test(t)#X-squared = 28.492, df = 3, p
chisq.test(u)#X-squared = 72.529, df = 4, p
chisq.test(v)#X-squared = 86.19, df = 8, p-
chisq.test(w) #X-squared = 17.505, df = 3,
chisq.test(x)#X-squared = 22.504, df = 3, p
chisq.test(y)#X-squared = 5.2411, df = 3, p
chisq.test(z)#X-squared = 0.00015475, df =
chisq.test(aa) #X-squared = 60.598, df = 3,
chisq.test(bb)#X-squared = 16.325, df = 3,
cramersV(m)
cramersV(n)
cramersV(o)
cramersV(p)
cramersV(q)
cramersV(r)
cramersV(s)
cramersV(t)
cramersV(u)
cramersV(v)
cramersV(w)
cramersV(x)
cramersV(y)
cramersV(z)
cramersV(aa)
cramersV(bb)
#Association with e/o
#Cramers V
library(lsr)
a<-table(raw$JobRole,raw$JobLevel)
b<-table(raw$JobRole,raw$Education)
c<-table(raw$JobRole,raw$Department)
d<-table(raw$JobRole,raw$EducationField)
e<-table(raw$JobRole,raw$MaritalStatus)
f<-table(raw$JobRole,raw$Gender)
g<-table(raw$JobLevel,raw$EducationField)
```

```
cramersV(f)
cramersV(g)
cramersV(h)
cramersV(i)
cramersV(j)
cramersV(k)
#Remove irrelevant categorical vars
raw$Gender<-NULL
raw$PerformanceRating<-NULL
raw$Education<-NULL
raw$RelationshipSatisfaction<-NULL
str(raw)
#visualization
cc<- scale_fill_brewer(palette="Reds")
#JE<- ggplot(raw,aes(Education,fill=JobRole))+geom_bar()+cc
JEF<- ggplot(raw,aes(EducationField,fill=JobRole))+geom_bar()+cc
#JG<- ggplot(raw,aes(JobRole,fill=Gender))+geom_bar()+cc
JJ<- ggplot(raw,aes(JobLevel,fill=JobRole))+geom_bar()+cc
JD<- ggplot(raw,aes(JobRole,fill=Department))+geom_bar()+cc
DEF<- ggplot(raw,aes(EducationField,fill=Department))+geom_bar()+cc
DJ<- ggplot(raw,aes(JobLevel,fill=Department))+geom_bar()+cc
#EFJ<- ggplot(raw,aes(JobLevel,fill=EducationField))+geom_bar()+cc
#JM<- ggplot(raw,aes(JobRole,fill=MaritalStatus))+geom_bar()+cc
MS<- ggplot(raw,aes(StockOptionLevel,fill=MaritalStatus))+geom_bar()+cc
#SJ<- ggplot(raw,aes(StockOptionLevel,fill=JobLevel))+geom_bar()+cc
grid.arrange(JEF,DEF,ncol=2,top = "Education Field Associations")
grid.arrange(JD,MS,ncol=2,top = "Other Associations")
grid.arrange(DJ,JJ,ncol=2,top = "Job Level Associations")
#Correlations
##3: Create an initial correlation plot to explore relationships
library(dplyr)
library(corpcor)
library(corrplot)
attach(raw)
shapiro.test(YearsInCurrentRole)
shapiro.test(YearsSinceLastPromotion)
shapiro.test(YearsWithCurrManager)
shapiro.test(YearsAtCompany)
shapiro.test(TotalWorkingYears)
shapiro.test(Age)
shapiro.test(DailyRate)
shapiro.test(MonthlyRate)
shapiro.test(MonthlyIncome)
shapiro.test(NumCompaniesWorked)
shapiro.test(DistanceFromHome)
raw.numeric <- select_if(raw, is.numeric)
```

```r
raw.numeric <- select_if(raw, is.numeric)
#raw.numeric<-subset(raw.numeric, select=-c(4,5,7,8,9,14:16,19))
colnames(raw.numeric)
correlations <-cor(raw.numeric)
corrplot(correlations, tl.srt=25,type="lower", method="number",
         tl.col="black", sig.level=0.05, insig="blank",tl.cex=.8,
         main="Numeric & Ordinal Correlations",mar=c(0,0,2,0))
#as.data.frame(correlations)
#Continuous Vars
##5: Explore relationships between highly correlated variables
library(caret)
set.seed(11)
Flagged <- findCorrelation(correlations,0.7, verbose=FALSE, names=TRUE,exact=TRUE)
print(Flagged) #Flagged Columns:"TotalWorkingYears","JobLevel","YearsAtCompany","Ye
#Association between continuous vars and target
att<-raw$Attrition
dd<-table(raw$Age,att)
nn<-table(raw$DailyRate,att)
oo<-table(raw$DistanceFromHome,att)
pp<-table(raw$HourlyRate,att)
qq<-table(raw$MonthlyRate,att)
rr<-table(raw$MonthlyIncome,att)
ss<-table(raw$NumCompaniesWorked,att)
tt<-table(raw$PercentSalaryHike,att)
uu<-table(raw$TotalWorkingYears,att)
vv<-table(raw$TrainingTimesLastYear,att)
ww<-table(raw$YearsInCurrentRole,att)
xx<-table(raw$YearsSinceLastPromotion,att)
yy<-table(raw$YearsWithCurrManager,att)
zz<-table(raw$YearsAtCompany,att)

chisq.test(dd)$p.value
chisq.test(nn)$p.value
chisq.test(oo)$p.value
chisq.test(pp)$p.value
chisq.test(qq)$p.value
chisq.test(rr)$p.value
chisq.test(ss)$p.value
chisq.test(tt)$p.value
chisq.test(uu)$p.value
chisq.test(vv)$p.value
chisq.test(ww)$p.value
chisq.test(xx)$p.value
chisq.test(yy)$p.value
chisq.test(zz)$p.value
```

```r
colnames(raw.numeric)
ggplot(raw,aes(MonthlyIncome,fill=Attrition))+geom_histogram(position="dodge")+ggtitle("Attrition by Income Level")
ggplot(raw,aes(JobLevel,fill=Attrition))+geom_histogram(position="dodge")+ggtitle("Attrition by Income Level")
#Remove redundant continuous vars
raw$DailyRate<-NULL
raw$MonthlyRate<-NULL
raw$YearsInCurrentRole<-NULL
raw$YearsWithCurrManager<-NULL
raw$MonthlyIncome<-NULL
raw.numeric2 <- select_if(raw, is.numeric)
cor2<-cor(raw.numeric2)
Flagged <- findCorrelation(cor2,0.7, verbose=FALSE, names=TRUE,exact=TRUE) #flag only correlations above the cutoff
print(Flagged)
str(raw)
#Feature Selection
library(randomForest)
library(caret)
library(nnet)
rawX<-raw[complete.cases(raw),]
p<-rawX
str(raw)
r.control <- trainControl(method="repeatedcv", number=5, repeats=3) #Variable Importance via Classification
set.seed(11)
set.seed(11)
importance.model.glm <- train(Attrition~., data=p, method="glm", trControl=raw.ctrl)
set.seed(11)
importance.model.svm <- train(Attrition~., data=p, method="svmRadial", trControl=raw.ctrl)
#Feature selection part 2
# var importance
redundant.glm <- varImp(importance.model.glm, scale=TRUE)
redundant.rf <- varImp(importance.model.rf, scale=TRUE)
c<-ggplot(redundant.glm,mapping=aes(redundant.glm))+ggtitle("GLM Feature Importance")
#d<-ggplot(redundant.svm,mapping=aes(redundant.svm))+ggtitle("SVM Feature Importance")
e<-ggplot(redundant.rf,mapping=aes(redundant.rf))+ggtitle("RF Feature Importance")#+geom_bar()#+scale_color_brewer(palette =
##6: Feature Selection
#raw.4<-subset(raw,select=c("Department","Attrition", "EnvironmentSatisfaction", "JobInvolvement","JobLevel", "JobSatisfacti
#RF FEATURES
raw.5<-subset(raw,select=c(Age,Attrition,TotalWorkingYears,HourlyRate,DistanceFromHome,
                           OverTime,YearsAtCompany,PercentSalaryHike,NumCompaniesWorked,EnvironmentSatisfaction,
                           JobLevel,StockOptionLevel,JobInvolvement,TrainingTimesLastYear,WorkLifeBalance,JobSatisfaction))
#svm features
#raw.6<-subset(raw,select=c("Age","Attrition","PercentSalaryHike","YearsSinceLastPromotion","DistanceFromHome","JobRole","Jo
colnames(raw)#dataset1
colnames(raw.5)#RF feature dataset
library(e1071)
library(DMwR)
```

```
295 library(rpart)
296 library(ROSE)
297 #Oversampling Trials
298 #SPLIT
299 split0<-createDataPartition(raw$Attrition,p = .75, list = FALSE) #raw
300 traw <- raw[split0, ] #training
301 sraw  <- raw[-split0, ] #test
302 set.seed(11)
303 split5 <- createDataPartition(raw.5$Attrition,p = .75, list = FALSE)
304 traw5 <- raw.5[split5, ] #training
305 sraw5  <- raw.5[-split5, ] #test
306 set.seed(11)
307 table(raw.5$Attrition)
308 table(traw5$Attrition)
309 #SMOTE
310 library(DMwR)
311 os0 <-SMOTE(Attrition ~ ., traw, perc.over = 100, perc.under=200)
312 #os4 <- SMOTE(Attrition ~ ., traw4, perc.over = 100, perc.under=200)
313 os5 <- SMOTE(Attrition ~ ., traw5, perc.over = 100, perc.under=200)
314 #os6 <- SMOTE(Attrition ~ ., traw6, perc.over = 100, perc.under=200)
315 prop.table(table(os0$Attrition))
316 prop.table(table(os5$Attrition))
317 #TRAIN
318 set.seed(11)
319 Train<- os0
320 Test<- sraw
321 set.seed(11)
322 fit.rf <- train(Attrition ~.,Train,
323                 method = 'rf', ntree = 2000,
324                 tuneGrid = data.frame(mtry = 6))
325 set.seed(11)
326 fit.glm <- train(Attrition ~.,Train,
327                  method = 'glm')
328 set.seed(11)
329 fit.svm <- train(Attrition~.,Train,
330                  method = 'svmRadial',
331                  cost=1,gamma=0.001)
332 set.seed(11)
333 fit.nn <- train(Attrition ~.,Train,
334                 method = 'pcaNNet')
335 set.seed(11)
336 fit.dt <- train(Attrition ~.,Train,
337                 method = 'rpart') #Decision Tree
338 print(fit.rf)
339
```

```r

#PREDICTIONS
#PREDICT
set.seed(11)
pred.rf <- predict(fit.rf, Test[,-2])
set.seed(11)
pred.nn <- predict(fit.nn, Test[,-2])
set.seed(11)
pred.glm <- predict(fit.glm, Test[,-2])
set.seed(11)
pred.svm <- predict(fit.svm,Test[,-2])
set.seed(11)
pred.dt <- predict(fit.dt,Test[,-2])
#gamma.range<-10^(-3:3)
#cost.range<-10^(-2:2)
#tuning<-tune.svm(Attrition~.,type="C-classification",
# gamma=gamma.range,cost=cost.range, data=os6)
#tuning$best.parameters
#tuning
summary(pred.rf)
#VALIDATE
b<-Test$Attrition
roc.curve(b,pred.glm,main="ROC curve: RF")
roc.curve(b,pred.svm,add.roc=TRUE, col=2)
roc.curve(b,pred.nn,add.roc=TRUE, col=3)
roc.curve(b,pred.rf,add.roc=TRUE, col=4)
roc.curve(b,pred.dt,add.roc=TRUE, col=5)
legend("bottomright", c("Logistic Regression", "Support Vector Machines", "Neural Network", "Random Forest","Decision Tree"),
       col=1:6, lty=1:6, lwd=2)

confusionMatrix(b,pred.glm)
confusionMatrix(b,pred.nn)
confusionMatrix(b,pred.rf)
confusionMatrix(b,pred.svm)
confusionMatrix(b,pred.dt)
```