HRAnalytics_Data_Analysis_PDF

Sikyun Lee

2020 10 5

This notebook is an analytics project to analyze the dataset provdied in the (IBM HR Analytics Employee Attrition & Performance)[https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset (https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset)]

The notebook will consist of Exploratory Data Analysis, Cohort Analysis by Attrition and Join Dates, and Prediction of Attrition.

Library Load

Data Import

```
data <- read.csv('WA_Fn-UseC_-HR-Employee-Attrition.csv')
#head(data)
length(colnames(data)) #35 columns
```

```
## [1] 35
```

Data Fix

There is an error reading in the Age column and this was fixed as below.

```
data_tbl <- as.tibble(data)
```

```
## Warning: `as.tibble()` is deprecated as of tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
#head(data_tbl)

`%>%` <- magrittr:: `%>%` #setting piping

#change a glitch in the Age column name
hr_data <- data_tbl %>%
  rename(Age = 癤풹ge)
```

Basic Data Check:

The following basic checks and others have been done including: - Number of Rows - Number of Distinct Employee information - Computations using dplyr functions - Cohort Analysis of Employment by the Years - Attrition by the Years in respect to Yearly Employment

```
#Data Processsing for Cohort Analysis

#Percentage of Attrition per JobRole
hr_data %>%
  group_by(Attrition, JobRole) %>%
  summarise(churn_employee_num = n()) %>%
  #filter(Attrition == "Yes") %>%
  arrange(desc(churn_employee_num)) %>%
  #filter(JobRole == "Sales Executive") %>%
  ungroup() %>%
  group_by(JobRole) %>%
  filter(Attrition == "Yes") %>%
  summarise(pct_attr = n()/churn_employee_num) %>%
  arrange(desc(pct_attr)) %>%
  glimpse #Yes: Top Laboratory Technician, Sales Executive, Research Scientist, Sales Representative, Huma
n Resources
```

```
## `summarise()` regrouping output by 'Attrition' (override with `.groups` argument)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## Rows: 9
## Columns: 2
## $ JobRole  <chr> "Research Director", "Manager", "Healthcare Representative...
## $ pct_attr <dbl> 0.50000000, 0.20000000, 0.11111111, 0.10000000, 0.08333333...
```

```
#No: Sales Executive, Research Scientist, Laboratory Technician, Manufacturing Director

#In terms of percentage per JobRole, Research Director 0.5, Manager 0.20000000, Healthcare Representative
 0.11111111, Manufacturing Director 0.10000000, Human Resources 0.08333333 are top roles and percentage th
at they leave compared to the total number of people

#Create cohort using YearsAtCompany, YearsInCurrentRole,YearsSinceLastPromotion, YearsWithCurrManager colu
mn
hr_data1 <- hr_data %>%
  mutate(date = as.Date(paste0(year(Sys.Date()), "-", month(Sys.Date()), "-01"))) %>%
  mutate(join_date = year(date) - YearsAtCompany) %>%
  mutate(curr_role_date = year(date) - YearsInCurrentRole) %>%
  mutate(last_promo = year(date) - YearsSinceLastPromotion) %>%
  mutate(curr_boss_date = year(date) - YearsWithCurrManager) %>%
  glimpse
```

```
## Rows: 1,470
## Columns: 40
## $ Age                     <int> 41, 49, 37, 33, 27, 32, 59, 30, 38, 36, 35...
## $ Attrition               <chr> "Yes", "No", "Yes", "No", "No", "No", "No"...
## $ BusinessTravel          <chr> "Travel_Rarely", "Travel_Frequently", "Tra...
## $ DailyRate               <int> 1102, 279, 1373, 1392, 591, 1005, 1324, 13...
## $ Department              <chr> "Sales", "Research & Development", "Resear...
## $ DistanceFromHome        <int> 1, 8, 2, 3, 2, 2, 3, 24, 23, 27, 16, 15, 2...
## $ Education               <int> 2, 1, 2, 4, 1, 2, 3, 1, 3, 3, 3, 2, 1, 2, ...
## $ EducationField          <chr> "Life Sciences", "Life Sciences", "Other",...
## $ EmployeeCount           <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ EmployeeNumber          <int> 1, 2, 4, 5, 7, 8, 10, 11, 12, 13, 14, 15, ...
## $ EnvironmentSatisfaction <int> 2, 3, 4, 4, 1, 4, 3, 4, 4, 3, 1, 4, 1, 2, ...
## $ Gender                  <chr> "Female", "Male", "Male", "Female", "Male"...
## $ HourlyRate              <int> 94, 61, 92, 56, 40, 79, 81, 67, 44, 94, 84...
## $ JobInvolvement          <int> 3, 2, 2, 3, 3, 3, 4, 3, 2, 3, 4, 2, 3, 3, ...
## $ JobLevel                <int> 2, 2, 1, 1, 1, 1, 1, 1, 3, 2, 1, 2, 1, 1, ...
## $ JobRole                 <chr> "Sales Executive", "Research Scientist", "...
## $ JobSatisfaction         <int> 4, 2, 3, 3, 2, 4, 1, 3, 3, 3, 2, 3, 3, 4, ...
## $ MaritalStatus           <chr> "Single", "Married", "Single", "Married", ...
## $ MonthlyIncome           <int> 5993, 5130, 2090, 2909, 3468, 3068, 2670, ...
## $ MonthlyRate             <int> 19479, 24907, 2396, 23159, 16632, 11864, 9...
## $ NumCompaniesWorked      <int> 8, 1, 6, 1, 9, 0, 4, 1, 0, 6, 0, 0, 1, 0, ...
## $ Over18                  <chr> "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y...
## $ OverTime                <chr> "Yes", "No", "Yes", "Yes", "No", "No", "Ye...
## $ PercentSalaryHike       <int> 11, 23, 15, 11, 12, 13, 20, 22, 21, 13, 13...
## $ PerformanceRating       <int> 3, 4, 3, 3, 3, 3, 4, 4, 4, 3, 3, 3, 3, 3, ...
## $ RelationshipSatisfaction <int> 1, 4, 2, 3, 4, 3, 1, 2, 2, 2, 3, 4, 4, 3, ...
## $ StandardHours           <int> 80, 80, 80, 80, 80, 80, 80, 80, 80, 80, 80...
## $ StockOptionLevel        <int> 0, 1, 0, 0, 1, 0, 3, 1, 0, 2, 1, 0, 1, 1, ...
## $ TotalWorkingYears       <int> 8, 10, 7, 8, 6, 8, 12, 1, 10, 17, 6, 10, 5...
## $ TrainingTimesLastYear   <int> 0, 3, 3, 3, 3, 2, 3, 2, 2, 3, 5, 3, 1, 2, ...
## $ WorkLifeBalance         <int> 1, 3, 3, 3, 3, 2, 2, 3, 3, 2, 3, 3, 2, 3, ...
## $ YearsAtCompany          <int> 6, 10, 0, 8, 2, 7, 1, 1, 9, 7, 5, 9, 5, 2,...
## $ YearsInCurrentRole      <int> 4, 7, 0, 7, 2, 7, 0, 0, 7, 7, 4, 5, 2, 2, ...
## $ YearsSinceLastPromotion <int> 0, 1, 0, 3, 2, 3, 0, 0, 1, 7, 0, 0, 4, 1, ...
## $ YearsWithCurrManager    <int> 5, 7, 0, 0, 2, 6, 0, 0, 8, 7, 3, 8, 3, 2, ...
## $ date                    <date> 2020-10-01, 2020-10-01, 2020-10-01, 2020-...
## $ join_date               <dbl> 2014, 2010, 2020, 2012, 2018, 2013, 2019, ...
## $ curr_role_date          <dbl> 2016, 2013, 2020, 2013, 2018, 2013, 2020, ...
## $ last_promo              <dbl> 2020, 2019, 2020, 2017, 2018, 2017, 2020, ...
## $ curr_boss_date          <dbl> 2015, 2013, 2020, 2020, 2018, 2014, 2020, ...
```

```
#Create a Attrition by Employees' joining Date
cohort_join_date <- hr_data1 %>%
  group_by(join_date) %>%
  summarise(join_year = n()) %>%
  filter(join_date >= 2014) %>%
  arrange(desc(join_date)) %>%
  glimpse
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## Rows: 7
## Columns: 2
## $ join_date <dbl> 2020, 2019, 2018, 2017, 2016, 2015, 2014
## $ join_year <int> 44, 171, 127, 128, 110, 196, 76
```

```
cohort_churn_join_date <- hr_data1 %>%
  filter(Attrition == "Yes") %>%
  group_by(join_date) %>%
  summarise(join_year = n()) %>%
  filter(join_date >= 2014) %>%
  arrange(desc(join_date)) %>%
  glimpse
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## Rows: 7
## Columns: 2
## $ join_date <dbl> 2020, 2019, 2018, 2017, 2016, 2015, 2014
## $ join_year <int> 16, 59, 27, 20, 19, 21, 9
```

```
hc <- highchart() %>%
  hc_chart(type = "column") %>%
  hc_title(text = "Annual New Employees") %>%
  #hc_add_series_list(ds) %>%
  hc_add_series(name = "Total Employees",
                type = 'column',
                showInLegend = FALSE,
                data = cohort_join_date,
                hcaes(name = join_date, y = join_year),
                lineWidth = 0,
                dataLabels = list(
                  enabled = TRUE,
                  formatter = highcharter::JS(
                        'function() {
                        if (this.y > 1000) {
                        return Highcharts.numberFormat((this.y/1000), 1) + "K"
                        } else {
                        return this.y
                        }
                        }'
                      )),
    #stacking = "no",
    enableMouseTracking = TRUE) %>%
  hc_add_series(name = "Total Attrition",
                type = 'column',
                showInLegend = FALSE,
                data = cohort_churn_join_date,
                hcaes(name = join_date, y = join_year),
                lineWidth = 0,
                dataLabels = list(
                  enabled = TRUE),
                #stacking = "no",
                enableMouseTracking = TRUE) %>%
  hc_xAxis(categories = unique(cohort_join_date$join_date)) %>%
  hc_yAxis(title = list(text = " " )) %>%
   hc_exporting(enables = TRUE,
                buttons = list(contextButton =
                                 list(menuItems = c('viewFullscreen',
                                                    'downloadPNG',
                                                    'separator',
                                                    'downloadCSV')))) %>%
  hc_add_theme(hc_theme_economist())
```

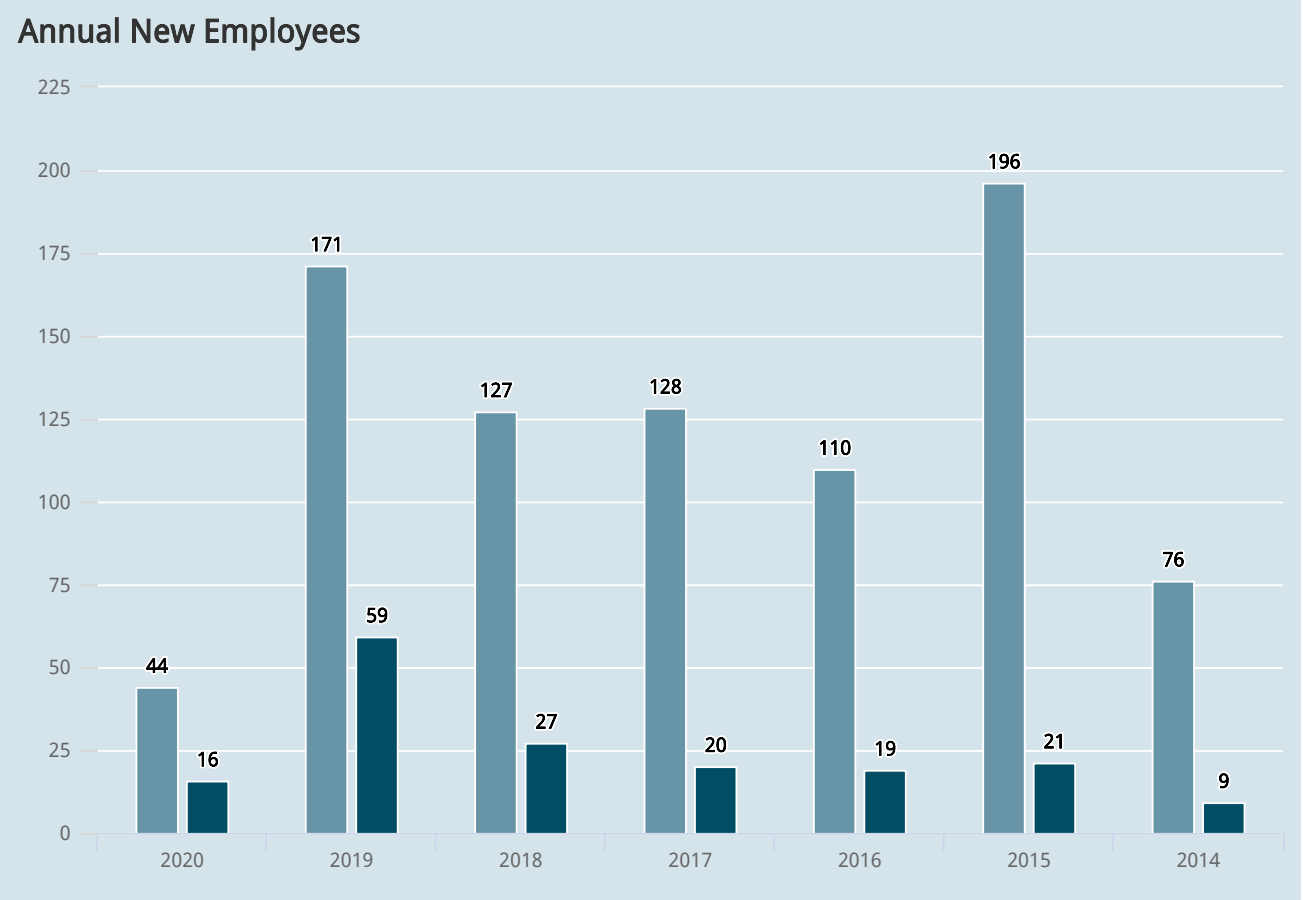
```
## Warning: `parse_quosure()` is deprecated as of rlang 0.2.0.
## Please use `parse_quo()` instead.
## This warning is displayed once per session.
```

```
## Warning: `group_by_()` is deprecated as of dplyr 0.7.0.
## Please use `group_by()` instead.
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
## Warning: `select_()` is deprecated as of dplyr 0.7.0.
## Please use `select()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
## Warning: `as_data_frame()` is deprecated as of tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

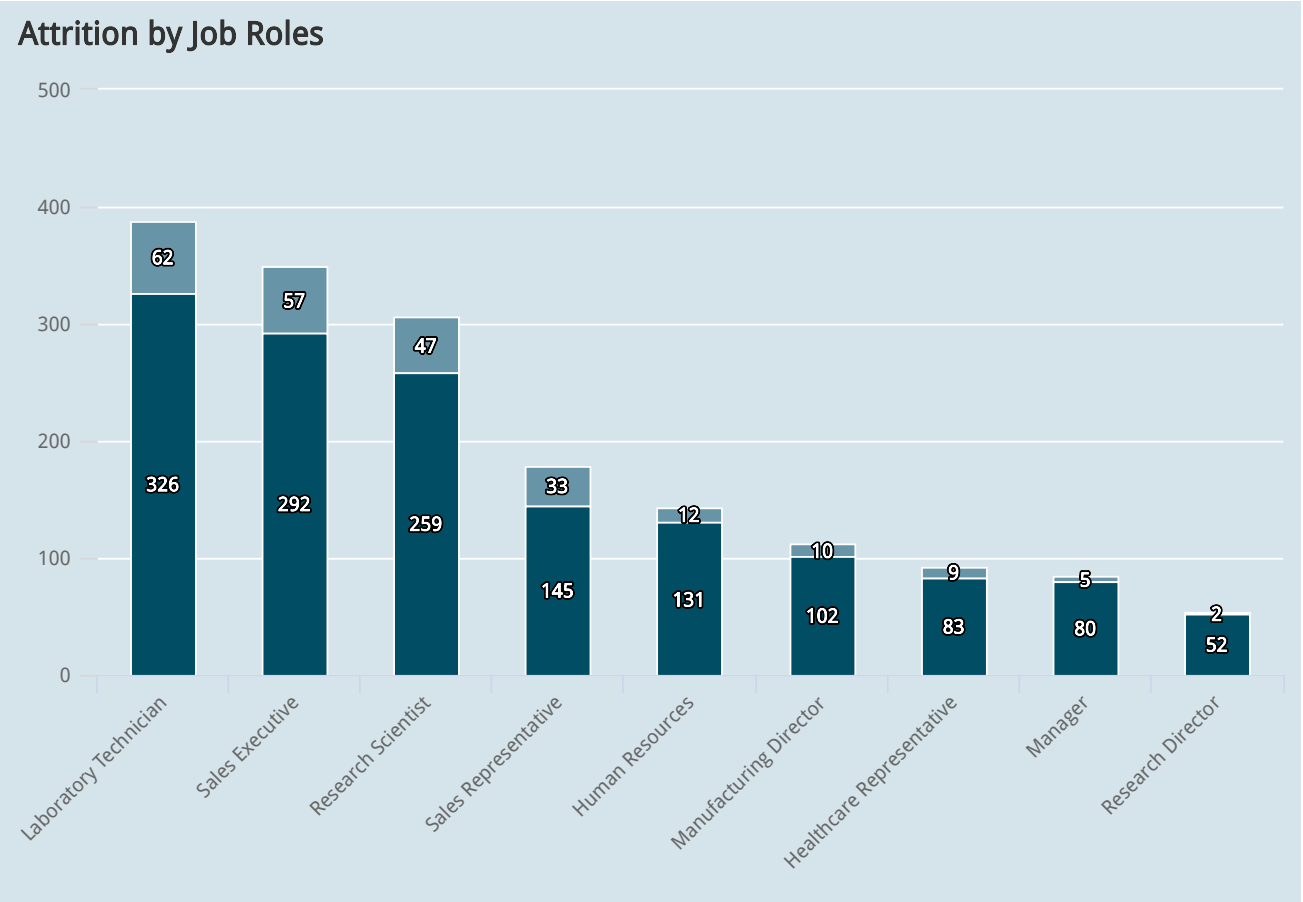
```
hc
```



First Exploratory Analysis Question: What is the yearly employment total and total churned (attrition)?

Based on years, 2019 seens the most attrition with the most employment while in terms of ratio, 2015 had the smallest attrition ratio in respect to employment.

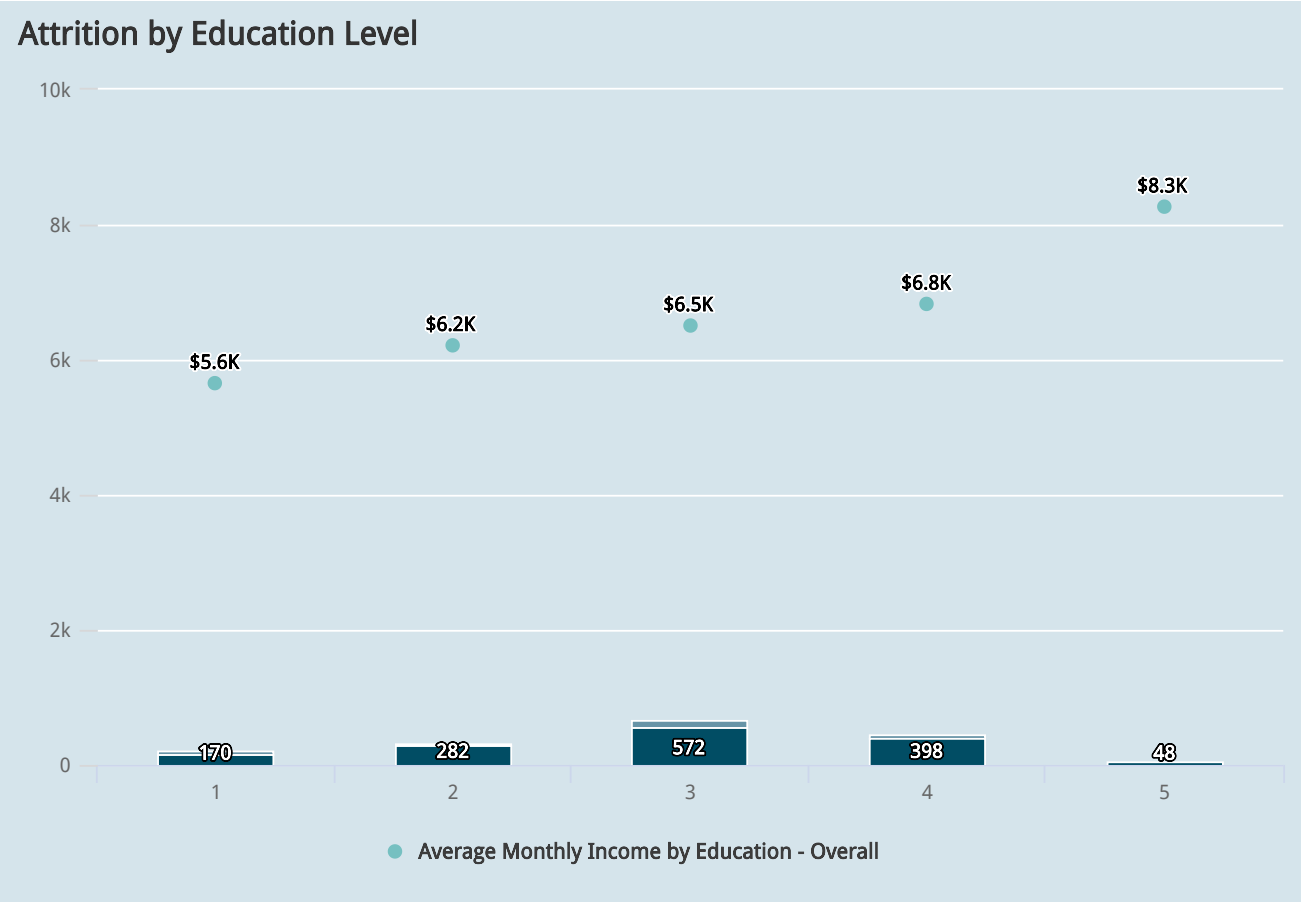
```
hc2 <- highchart() %>%
  hc_chart(type = "column") %>%
  hc_title(text = "Attrition by Job Roles") %>%
  #hc_add_series_list(ds) %>%
  hc_add_series(name = "Churns by Job Roles",
                type = 'column',
                showInLegend = FALSE,
                data = hr_data2,
                hcaes(name = JobRole, y = count),
                lineWidth = 0,
                dataLabels = list(
                  enabled = TRUE,
                  formatter = highcharter::JS(
                        'function() {
                        if (this.y > 1000) {
                        return Highcharts.numberFormat((this.y/1000), 1) + "K"
                        } else {
                        return this.y
                        }
                        }'
                      )),
    stacking = "normal",
    enableMouseTracking = TRUE) %>%
  hc_add_series(name = "Employees by Job Roles",
                type = 'column',
                showInLegend = FALSE,
                data = hr_data3,
                hcaes(name = JobRole, y = total),
                lineWidth = 0,
                dataLabels = list(
                  enabled = TRUE
                ),
                stacking = "normal",
                enableMouseTracking = TRUE) %>%
  hc_xAxis(categories = unique(hr_data2$JobRole)) %>%
  hc_yAxis(title = list(text = " " )) %>%
   hc_exporting(enables = TRUE,
                buttons = list(contextButton =
                                 list(menuItems = c('viewFullscreen',
                                                    'downloadPNG',
                                                    'separator',
                                                    'downloadCSV')))) %>%
  hc_add_theme(hc_theme_economist())
hc2
```



Second Exploratory Analysis Question: Which Job Roles have seen the highest number of churns (attritions)?

Based on this, Laboratory Technicians have the highest attrition followed by Sales Executive and Research Scientist.

```
hc3 <- highchart() %>%
  hc_chart(type = "column") %>%
  hc_title(text = "Attrition by Education Level") %>%
  #hc_add_series_list(ds) %>%
  hc_add_series(name = "Churns by Education Level",
                type = 'column',
                showInLegend = FALSE,
                data = hr_data5,
                hcaes(name = Education, y = count),
                lineWidth = 0,
                dataLabels = list(
                  enabled = TRUE,
                  formatter = highcharter::JS(
                        'function() {
                        if (this.y > 1000) {
                        return Highcharts.numberFormat((this.y/1000), 1) + "K"
                        } else {
                        return this.y
                        }
                        }'
                      )),
    stacking = "normal",
    enableMouseTracking = TRUE) %>%
  hc_add_series(name = "Employees by Education",
                type = 'column',
                showInLegend = FALSE,
                data = hr_data4,
                hcaes(name = Education, y = count),
                lineWidth = 0,
                dataLabels = list(
                  enabled = TRUE
                ),
                stacking = "normal",
                enableMouseTracking = TRUE) %>%
  hc_add_series(name = "Average Monthly Income by Education - Overall",
                type = 'line',
                showInLegend = TRUE,
                data = hr_data4,
                hcaes(name = Education, y = avg_incomerange),
                lineWidth = 0,
                dataLabels = list(
                  enabled = TRUE,
                  formatter = highcharter::JS(
                        'function() {
                        if (this.y > 1000) {
                        return "$" + Highcharts.numberFormat((this.y/1000), 1) + "K"
                        } else {
                        return this.y
                        }
                        }'
                      )),
                enableMouseTracking = TRUE) %>%
  hc_xAxis(categories = unique(hr_data5$Education)) %>%
  hc_yAxis(title = list(text = " " )) %>%
   hc_exporting(enables = TRUE,
                buttons = list(contextButton =
                               list(menuItems = c('viewFullscreen',
                                                  'downloadPNG',
                                                  'separator',
                                                  'downloadCSV')))) %>%
  hc_add_theme(hc_theme_economist())
hc3
```

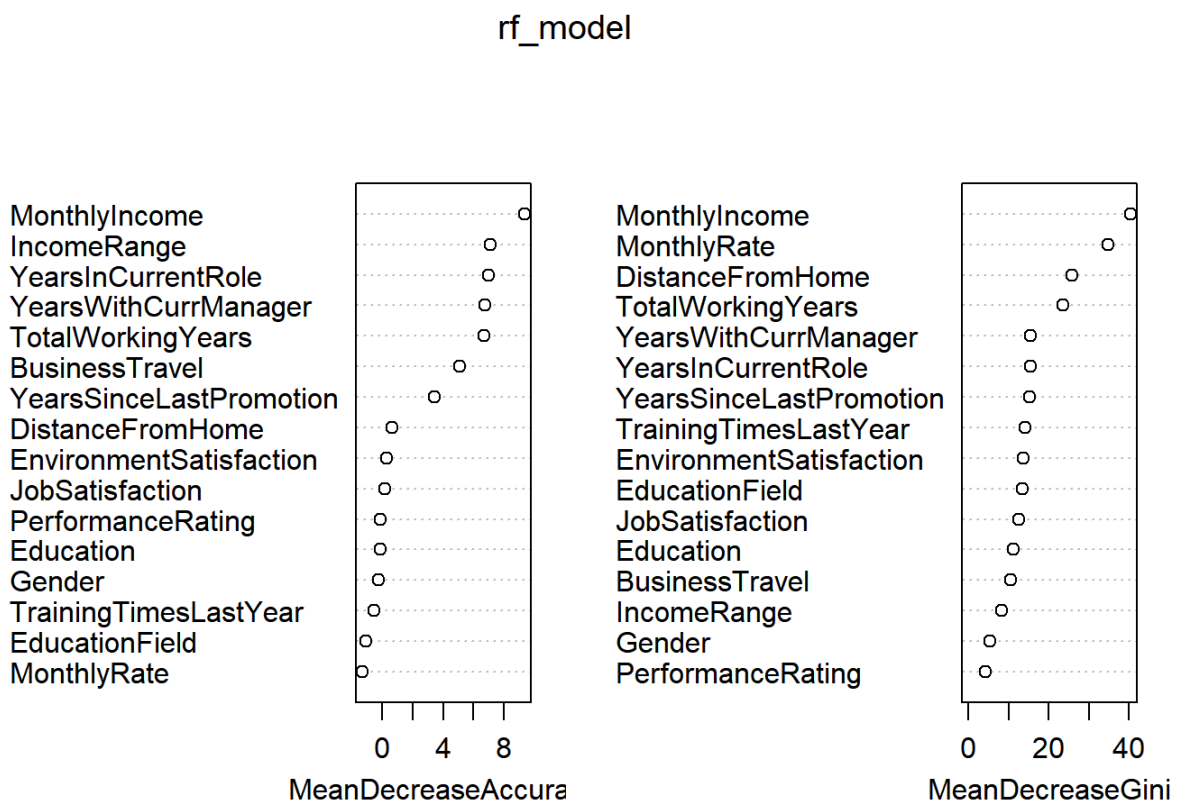


#Third Exploratory Analysis: What relationship does monthly income have with attrition and education? Based on the results, the most attrition is at level 3 eduation - where there are the most employees - and they have a “mid” level income. This is probably because these people have an opportunity to build a better career for themselves and pursue these better opportunities. This is evident based on the slight differences between the working environment satisfaction where the average Job Satisfaction - No is 2.78 while average Job Satisfaction - Yes is slightly lower at 2.47.

```
rf_data <- hr_data1
rf_data <- rf_data %>%
  mutate(Attrition = as.factor(Attrition)) %>%
  mutate(BusinessTravel = if_else(BusinessTravel == "Travel_Rarely", 1, if_else(BusinessTravel == "Travel_
Frequently", 2, 0))) %>%
  mutate(EducationField = if_else(EducationField == "Life Sciences", 1, if_else(EducationField == "Marketi
ng", 2, if_else(EducationField == "Technical Degree", 3, if_else(EducationField == "Medical", 4, if_else(E
ducationField == "Human Resources", 5, if_else(EducationField == "Other", 6, 0)))))) %>%
  mutate(IncomeRange = case_when(
  MonthlyIncome >= 0 & MonthlyIncome < 2000 ~ 1,
  MonthlyIncome >= 2000 & MonthlyIncome < 4000 ~ 2,
  MonthlyIncome >= 4000 & MonthlyIncome < 6000 ~ 3,
  MonthlyIncome >= 6000 & MonthlyIncome < 8000 ~ 4,
  MonthlyIncome >= 8000 ~ 5
)) %>%
  mutate(Gender = if_else(Gender == "Female", 2, 1)) %>%
  select(Attrition, BusinessTravel, Education, EducationField, IncomeRange, TotalWorkingYears, TrainingTim
esLastYear, YearsInCurrentRole, YearsSinceLastPromotion, YearsWithCurrManager, MonthlyIncome, MonthlyRate,
Gender, DistanceFromHome, JobSatisfaction, EnvironmentSatisfaction, PerformanceRating)

set.seed(110)
indexes <- sample(1:nrow(rf_data), size = 0.7*nrow(rf_data))
train_data <- rf_data[indexes,]
test_data <- rf_data[-indexes,]

rf_model <- randomForest(Attrition ~ ., train_data, importance = TRUE, ntree = 300)
result <- varImpPlot(rf_model)
```




```
rf_model
```

```
##
## Call:
##  randomForest(formula = Attrition ~ ., data = train_data, importance = TRUE,      ntree = 300)
##                Type of random forest: classification
##                      Number of trees: 300
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 15.16%
## Confusion matrix:
##      No Yes class.error
## No  864   8 0.009174312
## Yes 148   9 0.942675159
```

```
result
```

```
##                          MeanDecreaseAccuracy MeanDecreaseGini
## BusinessTravel                      5.1062135        10.581398
## Education                          -0.1746911        11.168020
## EducationField                     -1.0678788        13.347805
## IncomeRange                         7.0909123         8.182136
## TotalWorkingYears                   6.7043409        23.642239
## TrainingTimesLastYear              -0.5482600        14.182273
## YearsInCurrentRole                  6.9706300        15.495695
## YearsSinceLastPromotion             3.3886265        15.130876
## YearsWithCurrManager                6.7588212        15.566255
## MonthlyIncome                       9.3510007        40.333938
## MonthlyRate                        -1.3277516        34.801401
## Gender                             -0.2640225         5.394840
## DistanceFromHome                    0.6178794        25.814385
## JobSatisfaction                     0.1460487        12.644458
## EnvironmentSatisfaction             0.2903920        13.594088
## PerformanceRating                  -0.1368536         4.235408
```

Results from the Random Forest Model

Based on Random Forest model, Total Working Years, Monthly Income, MOntly Rate, Distance From Home, Years since Last Promotion are most related with Node Impurity, meaning they have the highest relation to the Attrition.

```
rf_pred <- predict(rf_model, newdata = test_data)
confusionMatrix(test_data$Attrition, rf_pred)
```

```
##      [,1] [,2]
## [1,]    0    0
## [2,]    0  361
```

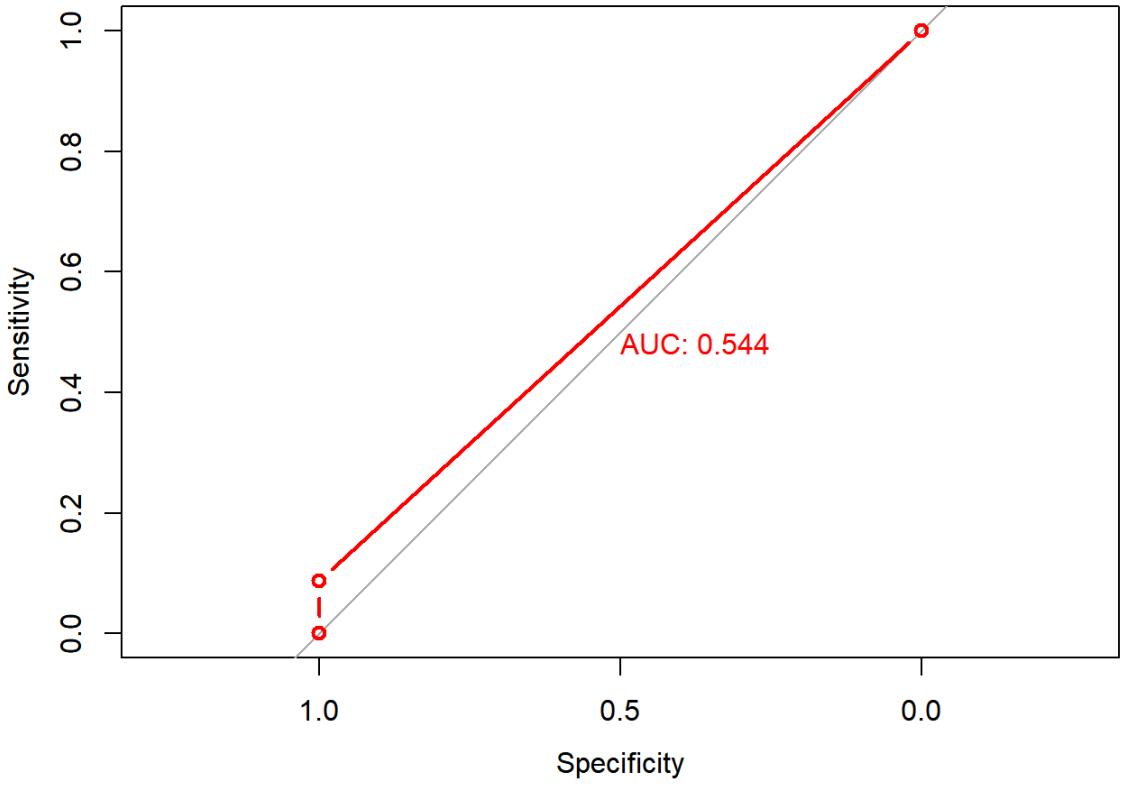
Evaluation on Random Forest Model

Based on the model, the accuracy of the model is approx. 83%.

```
rf_plot <- plot.roc(as.numeric(test_data$Attrition), as.numeric(rf_pred), lwd = 2, type ="b", print.auc = TRUE, col = "red")
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```



AUC on ROC

Although this is not a good fit or a model that explains the data well, this is a starting point in tuning the model by adding additional features and also better factoring the character data types. (this was a quick run so it didn't include all features)

Conclusion

A quick data exploration including simple aggregations and cohort analysis shows that employees in certain job roles, education level, and income level are employed more but are proned to churn. Though job satisfaction between those and those who did not churn are not large, this on the flip side shows that any employee can churn away depending on important factors such as Total Working Years, Monthly Income, MOntly Rate, Distance From Home, Years since Last Promotion. As a recommendation, although companies cannot realistically workout compensations to the level of every employee's satisfaction, they could work on distance from work - especially with work from home with COVID19 - and improve working conditions such as periodic trainings and opportunities to move up to a higher position in order to make employees feel that they are valued within the organization.