Contents

1.0 Introduction

1.1 Executive Summary

The COVID-19 epidemic and the "Great Resignation" phenomenon have significantly impacted the global labour market, changing work practices, the dynamics of the job market, and organizational work cultures. Due to a variety of factors, many employees have made the decision to voluntarily leave their jobs during the period, which has had a negative impact on organizations (Montaudon-Tomas et al., 2022). This report examined various models to best identify the predictive factors of employee churn. Subsequently, we will be conducting primary data exploratory and adopt the Random Forest algorithm using R Studio for performance analysis. From our findings, the critical attributes of employee churn ranked by importance are Monthly Income, Overtime, Age, Total Experience and Daily Rate, with an in-depth review documented in the respective section.

1.2 Background

The COVID-19 epidemic has significantly disrupted the worldwide labour market and changed people's work habits and the nature of the employment market in ways that have never been observed before. In addition, the "Great Resignation," which defines a pattern of workers willingly leaving their employment in significant numbers, is another remarkable phenomenon that has evolved alongside the epidemic. Together, these two variables have caused significant changes in the workforce, which have impacted the dynamics of the labour market and numerous elements of organizational work culture.

Many employees quit their employment during the COVID-19 epidemic and the Great Resignation due to numerous reasons that altered their work choices and expectations. The adverse outcomes may come in many forms. First, a high employee turnover rate can undermine organizational stability and result in lower productivity (Booth & Zoega, 1999), more work for the remaining staff, and more significant hiring expenses. Second, the need for more knowledgeable workers might reduce institutional knowledge, which could impact the standard of the job and client satisfaction. Abbasi and Hollman (2000) further added that employee resignations can also negatively affect the team's morale and make the remaining staff members stressed and anxious since they could be worried about their future employment. Consequently, in this report, we will examine various algorithms to identify which ones can best predict the traits of possible resigning employees to lower staff turnover, employment costs, and knowledge loss.

2.0 Data Exploratory

Job Sentiment

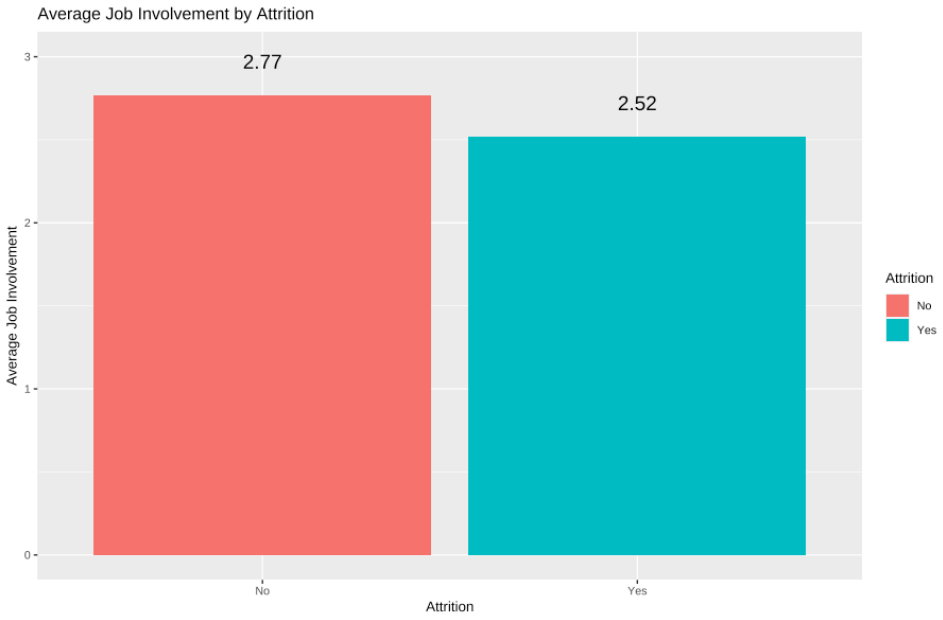

Figure 1: Job Involvement by Attrition

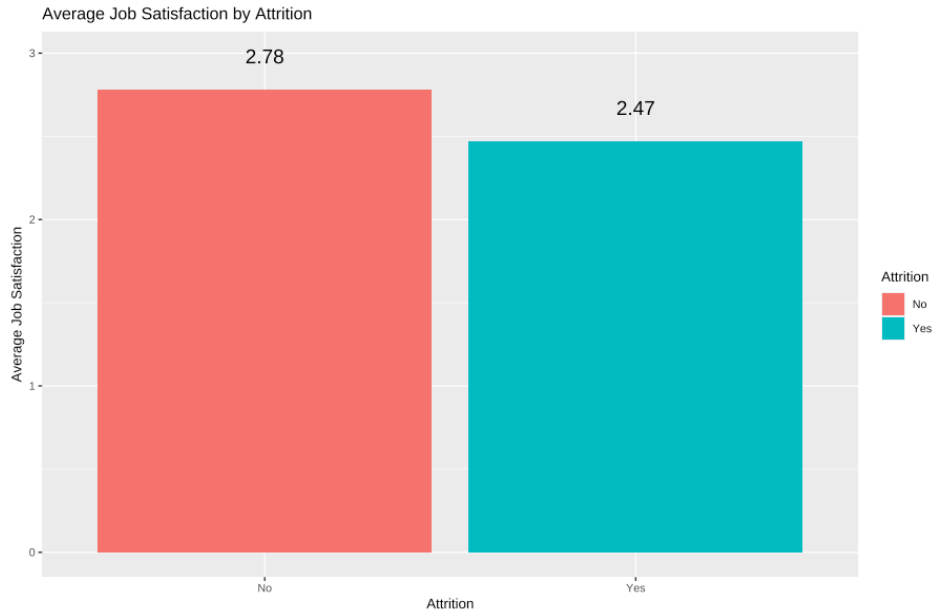

Figure 2: Job Satisfaction by Attrition

From Figures 1 and 2, we can observe that lower involvement and satisfaction lead to employee attrition. Interestingly, both graphs present similar values, with a segregation threshold of above 2.7 for employees that are likely to stay.

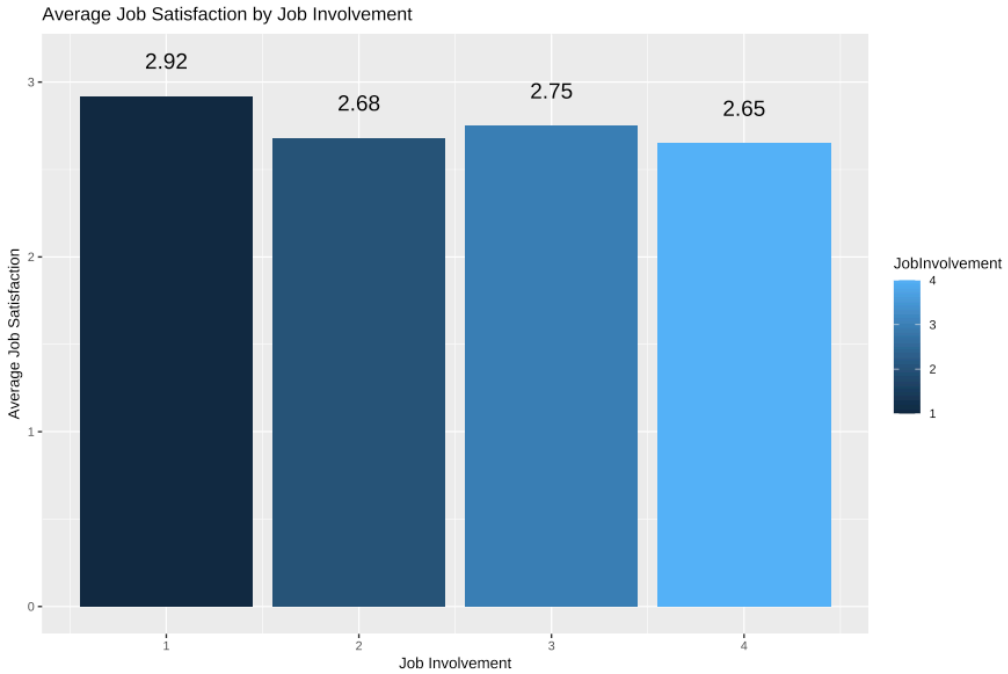

Figure 3: Job Satisfaction by Involvement

As mentioned, employees with a threshold of 2.7 are likely to be retained. Therefore, Group 1 and 3 are deemed to be less prone to attrition. On the contrary, Group 2 and 4 are more likely to experience a shorter tenure which requires further investigation.

Department Breakdown

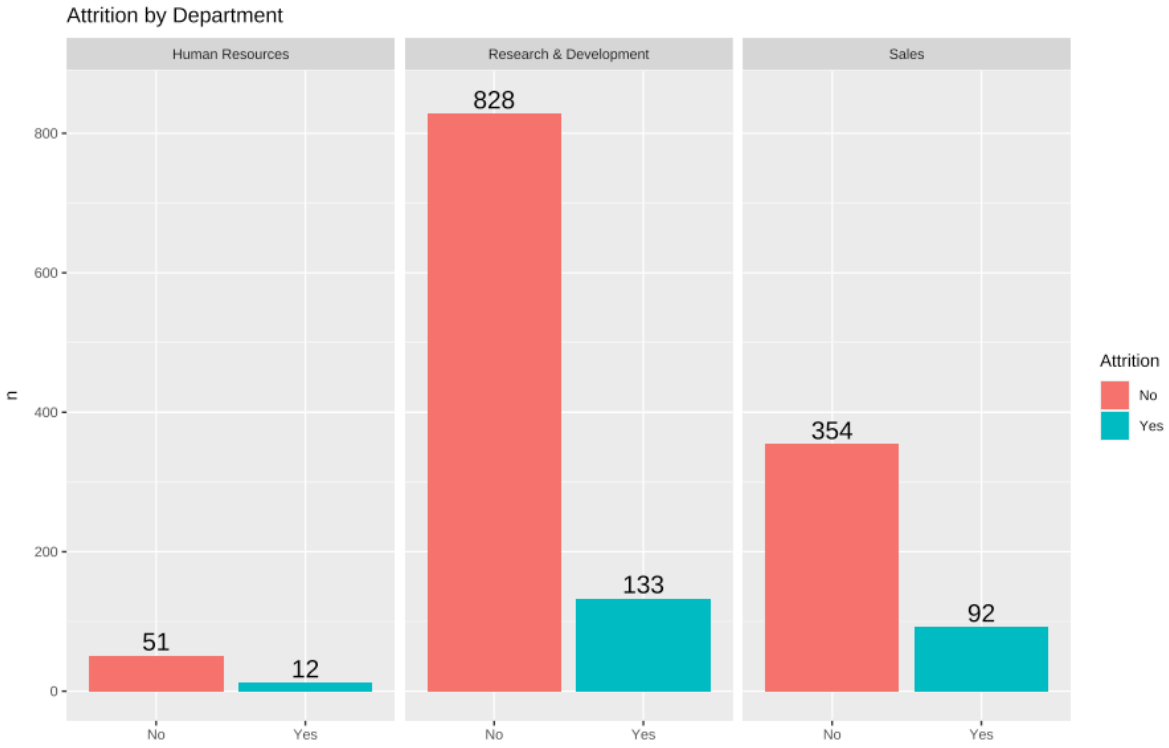

Figure 4: Department Attrition

At a glance, the Research and Development Department has the highest attrition figure. The substantial turnover could indicate underlying factors. However, scaled to the ratio for a better comparison, the Sales Department have the highest attrition rate of 21%, which might not be noticeable immediately. Therefore, more resources could be invested into retaining valuable employees of the Sales Department as it is the revenue-generating centre of any organisation.

Work-Life Balance

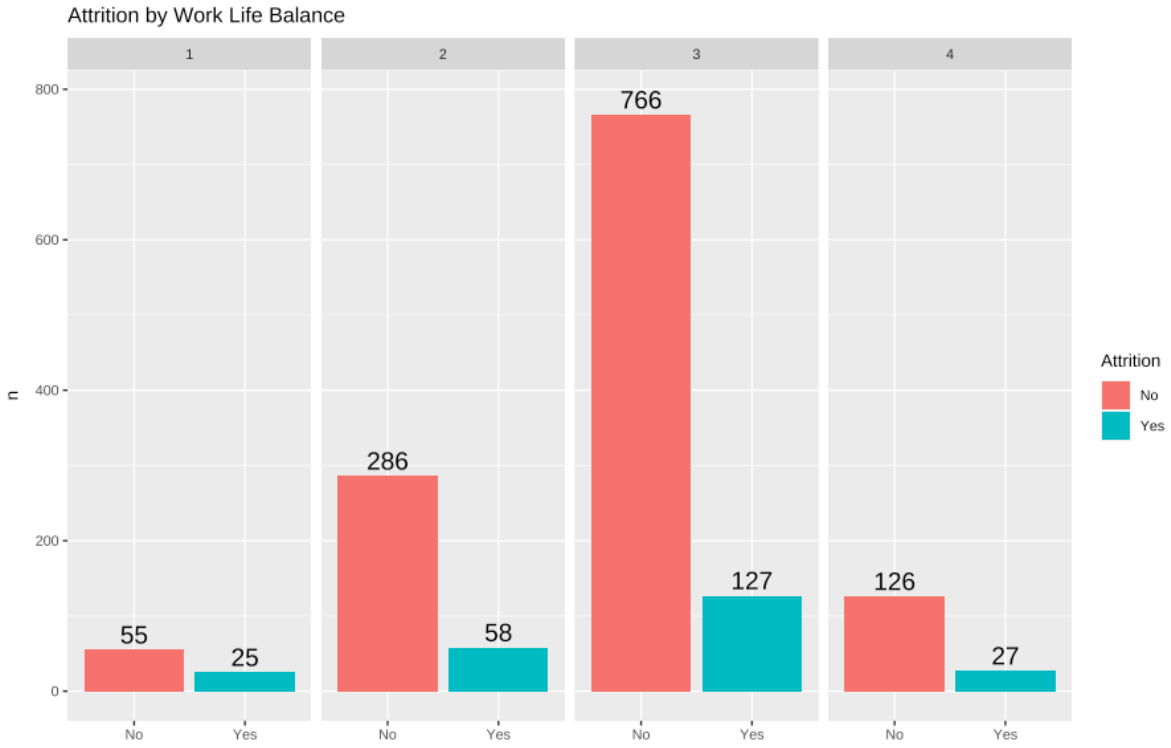

Figure 5: Attrition by Work-Life Balance

Similarly, Group 3 is deemed to have the most significant attrition by count. However, Group 3 has the lowest attrition ratio at 14%. Unsuspectingly, Group 1 has the highest attrition rate of 31% and require further analysis as it has double the rate of Group 3.

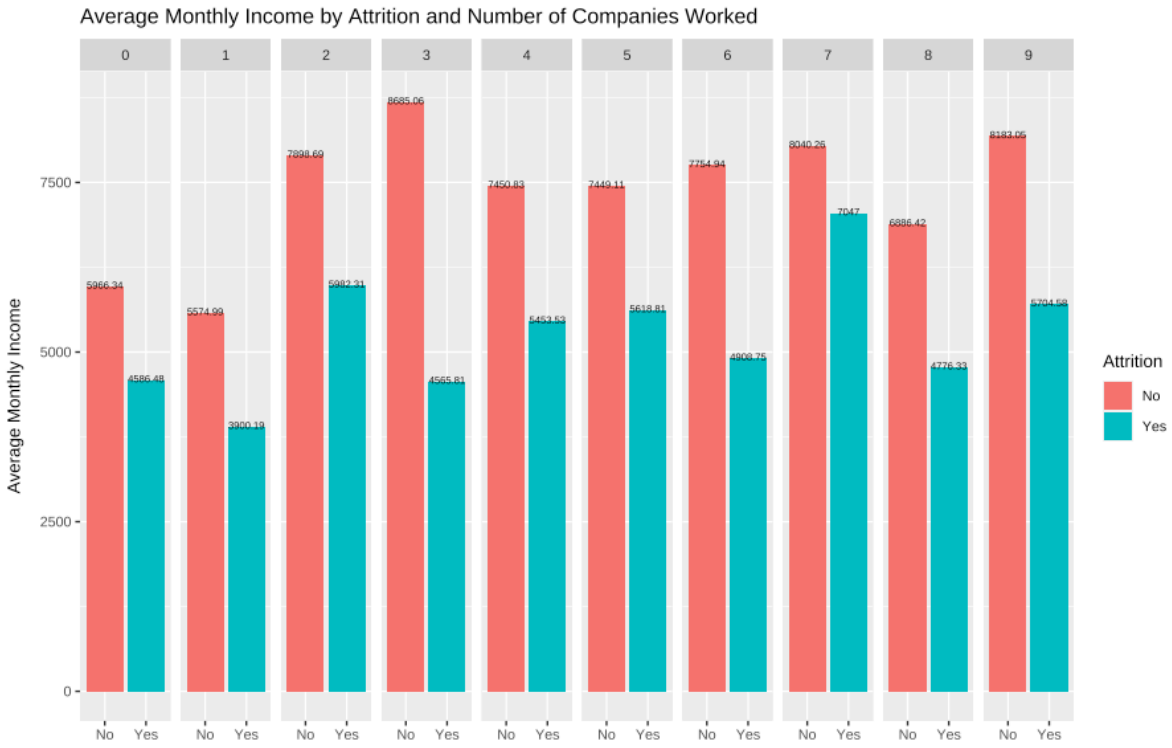

Figure 6: Breakdown of attrition by Salary and No. of Companies

According to the graph, employees who leave the firm usually have the lowest monthly income, regardless of the number of companies they work for. This trend demonstrates that monthly salary is among the top critical factors in employee retention. Furthermore, there is a tendency for employees who have worked for 2 to 9 companies to earn a more significant compensation. This behaviour can be explained by job hoppers or employees that have previously worked at a limited number of companies, e.g. fresh graduates.

Work Relations

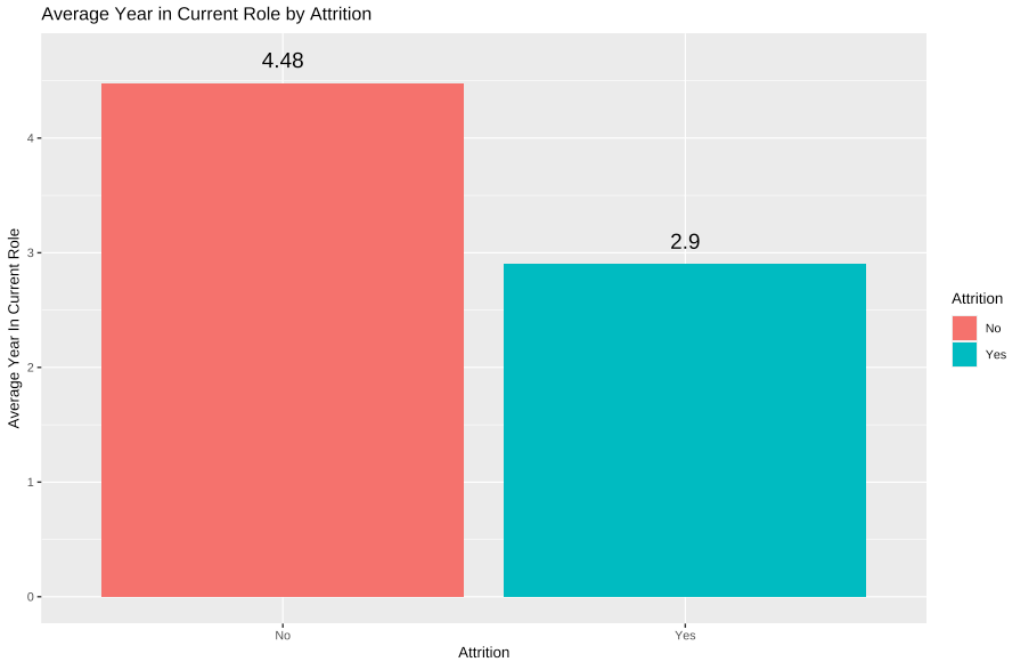

Figure 7: Time Spent in Current Role

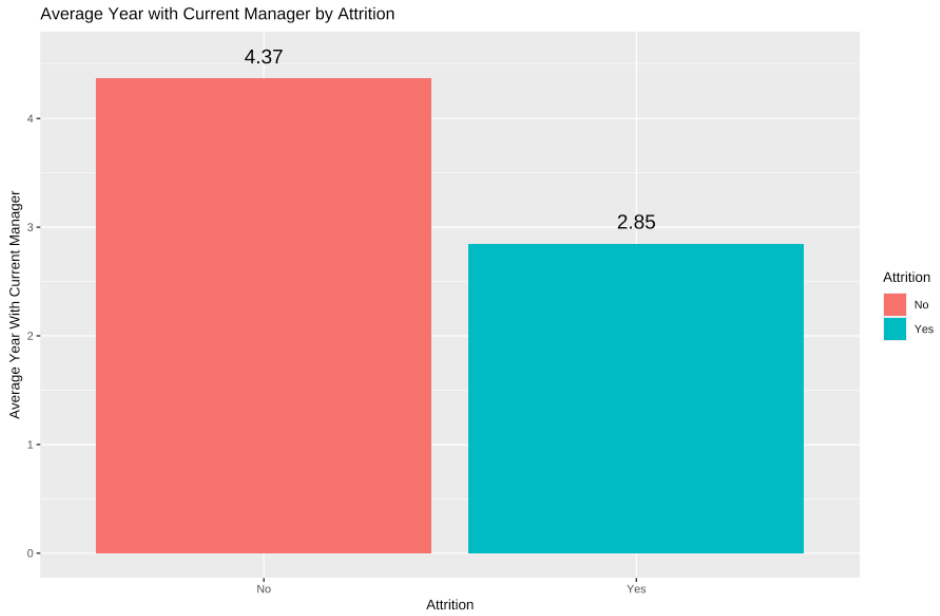

Figure 8: Time Spent with Manager

In general, a shorter tenure in the current role or time with the manager is associated with attrition. This could be due to cultural misfits or relations between employees and managers, impacting tenure.

3.0 Data Understanding & Preparation

3.1 Diving into the Data

Link: https://www.kaggle.com/datasets/thedevastator/employee-attrition-and-factors?select=HR_Analytics.csv.csv

The Employee Attrition and Factors dataset was extracted from Kaggle, which contains the employee attribution result (dependent variable) and the (independent) factors. The dataset has 1470 rows excluding headers, comprising 34 factors and the attrition outcome. These factors include demographic information about the employee (e.g. age), employee sentiments (e.g. Satisfaction, Work-Life Balance), job-related factors and financials (e.g. Tenure, Department). With this information, we can derive insights about employee churn from multiple angles and further analyse the cause of employment churn.

3.2 Data Preparation

Data integrity checks and transformations are performed upon obtaining the dataset to ensure that the data quality would not affect the subsequent data exploratory and analysis. Subsequently, the NA.omit functionality is done to scan for missing data. This would ensure that incomplete data are removed and is more effective than manual verification, which could be time-consuming and error-prone.

Omit NA

```
# Cleaning the data
data <- na.omit(data)
```

Figure 9: Removal of Missing Data Code Snippet

After the data integrity is ensured, data engineering is performed to encode the data (e,g. Attrition, Business Travel). The string data are automatically indexed based on the alphanumeric sequence. This process is known as label encoding. Because decision trees function by periodically separating the data based on the values of the input variables, they require numeric input data. When non-numeric variables, such as category or text, are included in the input data, decision trees cannot complete the necessary computations to identify the appropriate split.

Label Encoding

```
# Preprocess the data
# Transform the variable from chr data type to factor
data <- data %>% mutate_if(is.character,as.factor)
```

```r
# Attrition
levels(data$Attrition) <- seq(length(levels(data$Attrition)))
data$Attrition <- as.numeric(data$Attrition)

# BusinessTravel
levels(data$BusinessTravel) <- seq(length(levels(data$BusinessTravel)))
data$BusinessTravel <- as.numeric(data$BusinessTravel)

# Department
levels(data$Department) <- seq(length(levels(data$Department)))
data$Department <- as.numeric(data$Department)

# EducationField
levels(data$EducationField) <- seq(length(levels(data$EducationField)))
data$EducationField <- as.numeric(data$EducationField)

# Gender
levels(data$Gender) <- seq(length(levels(data$Gender)))
data$Gender <- as.numeric(data$Gender)

# JobRole
levels(data$JobRole) <- seq(length(levels(data$JobRole)))
data$JobRole <- as.numeric(data$JobRole)

# MaritalStatus
levels(data$MaritalStatus) <- seq(length(levels(data$MaritalStatus)))
data$MaritalStatus <- as.numeric(data$MaritalStatus)

# Over18
levels(data$Over18) <- seq(length(levels(data$Over18)))
data$Over18 <- as.numeric(data$Over18)

# OverTime
levels(data$OverTime) <- seq(length(levels(data$OverTime)))
data$OverTime <- as.numeric(data$OverTime)
```

Figure 10: Data Encoding Snippet

Non-correlated variables may introduce noise and unnecessary complexity. In order to further enhance the performance of the model, non-correlated variables are removed. This is done by calculating the variable correlation to attrition using the correlation function. This would increase the model's accuracy, reducing overfitting and complexity with reduced noise. Furthermore, this would hasten model inference and training while improving model interpretability.

Calculate Correlation

```
# Calculate the variable correlation value with Attrition
correlation_table <- cor(data %>% select(Attrition, everything())) %>%
as.data.frame()
correlation_table <- correlation_table[, 1, drop=FALSE] %>% arrange(-Attrition)
```

Figure 11: Correlation Function Snippet

The following output are as follows in the correlation table generated:

Variables	Attrition
OverTime	0.25
MaritalStatus	0.16
DistanceFromHome	0.08
JobRole	0.07
Department	0.06
NumCompaniesWorked	0.04
Gender	0.03
EducationField	0.03
MonthlyRate	0.02
PerformanceRating	0.00
BusinessTravel	0.00
HourlyRate	-0.01
EmployeeNumber	-0.01
PercentSalaryHike	-0.01
Education	-0.03
YearsSinceLastPromotion	-0.03
RelationshipSatisfaction	-0.05
DailyRate	-0.06
TrainingTimesLastYear	-0.06
WorkLifeBalance	-0.06

EnvironmentSatisfaction	-0.10
JobSatisfaction	-0.10
JobInvolvement	-0.13
YearsAtCompany	-0.13
StockOptionLevel	-0.14
YearsWithCurrManager	-0.16
Age	-0.16
MonthlyIncome	-0.16
YearsInCurrentRole	-0.16
JobLevel	-0.17
TotalWorkingYears	-0.17
EmployeeCount	*NA*
Over18	*NA*
StandardHours	*NA*
Showing 1 to 29 of 35 entries, 1 total columns	

Figure 12: Correlation Table

From the table above, Employee Count, Over 18, and StandardHours are not correlated to Attribution and therefore would be removed.

Drop the features with NA correlation

```
# Drop the column without correlation
data <- data[, -which(names(data) == "EmployeeCount")]
data <- data[, -which(names(data) == "Over18")]
data <- data[, -which(names(data) == "StandardHours")]
```

Figure 13: Variable Removal Snippet

Lastly, incorporating the n-fold cross validation into the models to obtain more reliable and robust performance results.

N-Fold Cross Validation

```
# Logistic Regression
# Train the model using N-Fold Cross Validation
control <- trainControl(method="cv", number=5)
lr <- train(Attrition~., data=trainData, method="glm", trControl=control)

...
```

```
# Random Forest
# Train the model using N-Fold Cross Validation
control <- trainControl(method = "cv", number = 5)
rf <- train(Attrition~., data=trainData, method="rf", trControl=control,
tuneLength=3, ntree=750, type='classification')
```

Figure 14: N-Fold Cross Validation Snippet

Including n-fold cross-validation in machine learning models can produce more robust and dependable outcomes. Variables like dataset size, model complexity, and processing resources determine the value of n. Five-fold cross-validation is frequently recommended because it balances successful model training and testing with the computational economy. It has been demonstrated that it has low bias and moderate variation in predicting model performance, making it a viable choice for many machine-learning applications. On the other hand, the best value of n will vary based on the individual problem being addressed and should be established on a case-by-case basis.

4.0 Algorithms Selection & R Implementation

4.1 Logistic Regression

Supervised learning algorithms such as Logistic Regression model relationships between the dependent and independent variables for binary or multicategory classification. Analyzing the coefficients or weights related to each feature allows for the use of Logistic Regression to identify characteristics and determines the likelihood of an event by analyzing the relationship between the variable(s). (IBM, n.d.).

Logistic Regression and Linear Regression are very similar except for the algorithm approach and application. Logistic Regression is used to classify data into binary or multiple classes, whereas Linear Regression is used to predict continuous numerical values. Furthermore, Logistic Regression is used to address classification issues, while regression issues are addressed by Linear Regression. (*Logistic Regression in Machine Learning*, n.d.). Therefore, Logistic Regression would better suit our employee attrition analysis that has a binary result (Attrition: Yes, No).

However, one of its drawbacks is that it assumes that features and outcomes always have linear associations, which may not be valid for complex data with nonlinear interactions. Additionally, it does not record feature interactions, which are crucial for several tasks involving the identification of attributes. Furthermore, Logistic Regression may have trouble with big or high-dimensional datasets because it is susceptible to overfitting.

Logistic Regression

```
# Train the model using N-Fold Cross Validation
control <- trainControl(method="cv", number=5)
lr <- train(Attrition~., data=trainData, method="glm", trControl=control)

# Test the model
lrPredResult <- predict(lr,newdata=testData)
lrPredResult <- ifelse(lrPredResult>1.5,2,1)
lrPredResult %>% head
lrPredResultDf <-
confusionMatrix(factor(testData$Attrition),factor(lrPredResult))
```

Figure 15: Logistic Regression Model Snippet

Logistic regression model performance result:

#	Performance	LR
1	Accuracy	0.8197
2	95% CI	(0.7709 , 0.862)
3	Kappa	0.1265
4	FI	0.899
5	MSE	0.1803
6	MAE	0.1803
7	Precision Rate	0.9958
8	Recall Rate	0.8194

Showing 1 to 8 of 8 entries, 2 total columns

Figure 16: Logistic Regression Model Performance

4.2 Random Forest Decision Tree

Random forest is a powerful predictive modelling technique for massive, complex datasets. It is an algorithm that is built upon the decision tree. As its name indicates, a random forest is a collection of various independent decision trees that function as an ensemble (Yiu, 2019). The final forecast is the average of all forecasts given by all decision trees, and this procedure is repeated several times.

Ensemble models offer more accurate predictions due to their propensity to have fewer biases and their capacity to lessen overfitting by merging several separate individual models. Additionally, it can manage both categorical and continuous data.

Random Forest

```
# Random Forest - 500
```

```
# Train the model using N-Fold Cross Validation
control <- trainControl(method = "cv", number = 5)
rf <- train(Attrition~., data=trainData, method="rf", trControl=control,
tuneLength=3, ntree=500, type='classification')
# Test the model
rfPredResult <- predict(rf,newdata=testData)
rfPredResult <- ifelse(rfPredResult>1.5,2,1)
rfPredResult %>% head
rfPredResultDf500 <- confusionMatrix(factor(testData$Attrition),
factor(rfPredResult))

# Random Forest - 750
# Train the model using N-Fold Cross Validation
control <- trainControl(method = "cv", number = 5)
rf <- train(Attrition~., data=trainData, method="rf", trControl=control,
tuneLength=3, ntree=750, type='classification')
# Test the model
rfPredResult <- predict(rf,newdata=testData)
rfPredResult <- ifelse(rfPredResult>1.5,2,1)
rfPredResult %>% head
rfPredResultDf750 <- confusionMatrix(factor(testData$Attrition),
factor(rfPredResult))

# Random Forest - 1000
# Train the model using N-Fold Cross Validation
set.seed(110)
control <- trainControl(method = "cv", number = 5)
rf <- train(Attrition~., data=trainData, method="rf", trControl=control,
tuneLength=3, ntree=1000, type='classification')
# Test the model
rfPredResult <- predict(rf,newdata=testData)
rfPredResult <- ifelse(rfPredResult>1.5,2,1)
rfPredResult %>% head
rfPredResultDf1000 <- confusionMatrix(factor(testData$Attrition),
factor(rfPredResult))
```

Figure 17: Random Forest Model Snippet

Random forest model's Performance Result:

#	Performance	RF500	RF750	RF1000
1	Accuracy	0.8163	0.8231	0.8197
2	95% CI	(0.7672 , 0.8589)	(0.7746 , 0.865)	(0.7709 , 0.862)
3	Kappa	0.1687	0.1841	0.1763

4	FI	0.8958	0.9	0.8979
5	MSE	0.1837	0.1769	0.1803
6	MAE	0.1837	0.1769	0.1803
7	Precision Rate	0.9789	0.9873	0.9831
8	Recall Rate	0.8256	0.8269	0.8262
Showing 1 to 8 of 8 entries, 4 total columns				

Figure 18: Random Forest Model Performance

The default 500 and 1000 branches, along with 750 branches were tested for comparison purposes. From the above, we can observe that the model with 750 branches performs the best among the three. The critical consideration is to ensure that the model can perform accurately and identify "True Positives". We tested by scaling along 50(s) and 100(s) branches between 500 and 1000, before deriving at 750 branches which has the optimal result. This is to ensure that we can accurately depict the traits of churned employees. Therefore, our key parametric would be Accuracy, F1, Precision and Recall Rate.

4.3 Model Selection

Performance comparison result:

#	**Performance**	**LR**	**RF750**
1	Accuracy	0.8197	0.8231
2	95% CI	(0.7709 , 0.862)	(0.7746 , 0.865)
3	Kappa	0.1265	0.1841
4	FI	0.899	0.9
5	MSE	0.1803	0.1769
6	MAE	0.1803	0.1769
7	Precision Rate	0.9958	0.9873
8	Recall Rate	0.8194	0.8269
Showing 1 to 8 of 8 entries, 4 total columns			

Figure 19: Model Performance Comparison

Based on the result above, Random Forest performed better in every aspect of the performance measurements in our test. This indicates that the Random Forest with 750 branches can achieve better accuracy and identify true positives. Unlike Logistic Regression, Random Forest does not assume linearity and can capture complex non-linear relationships between the features. Furthermore, Logistic Regression is prone to overfitting when the number of features is large or complex and may require domain knowledge for selecting features. On the contrary, Random Forest uses ensemble learning. It is able to identify essential features derived from the calculated importance automatically.

However, Random Forest is more complex than Logistic Regression. This would insinuate that Random Forest is more resource intensive and less easily interpretable than Logistic Regression.

5.0 Data Analysis

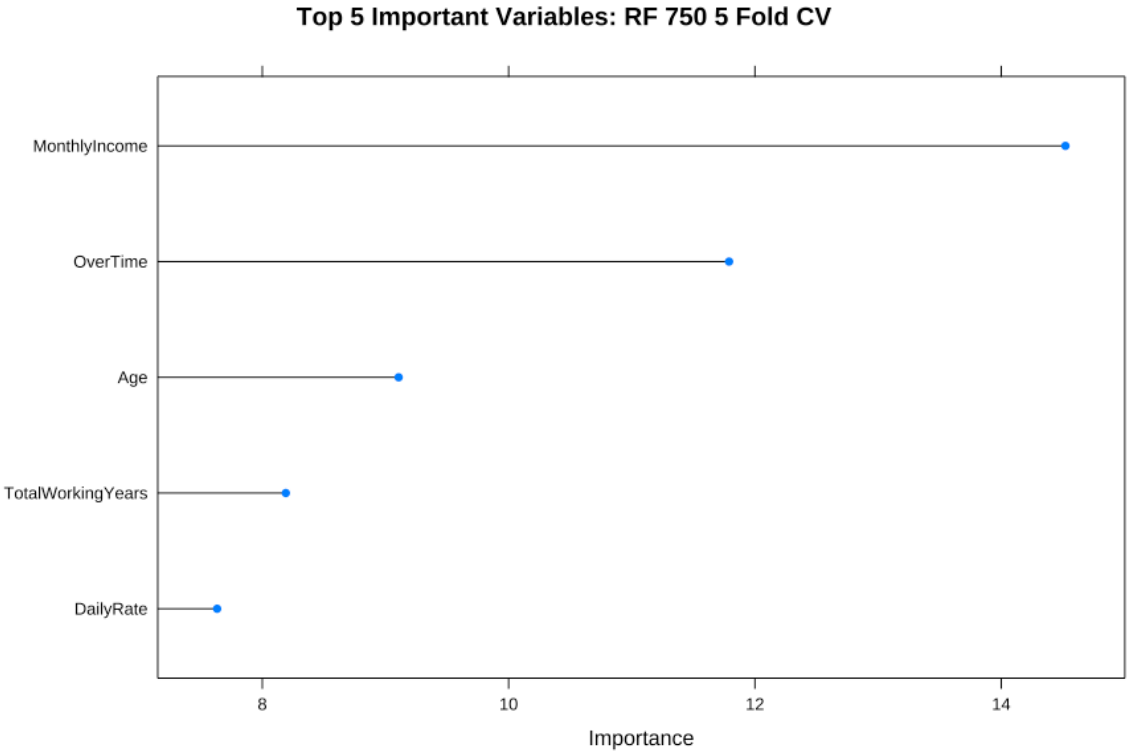

Figure 20: Top 5 RF750 Attrition Variables

After putting the Random Forest through training, the top 5 variables identified by the Random Forest models are as follows:

1. Monthly Income
2. Over Time
3. Age
4. Total Working Years
5. Daily Rate

Top 1st - Monthly Income

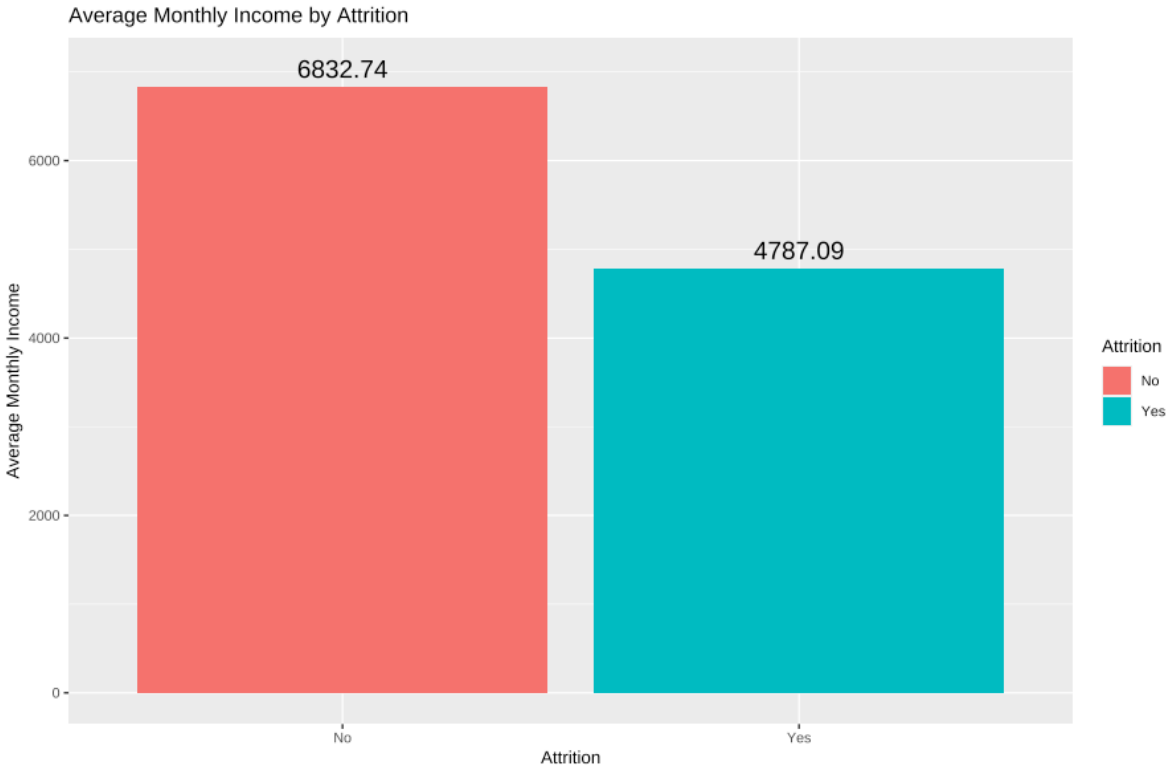

Figure 21: Monthly Income Analysis (Bar Chart)

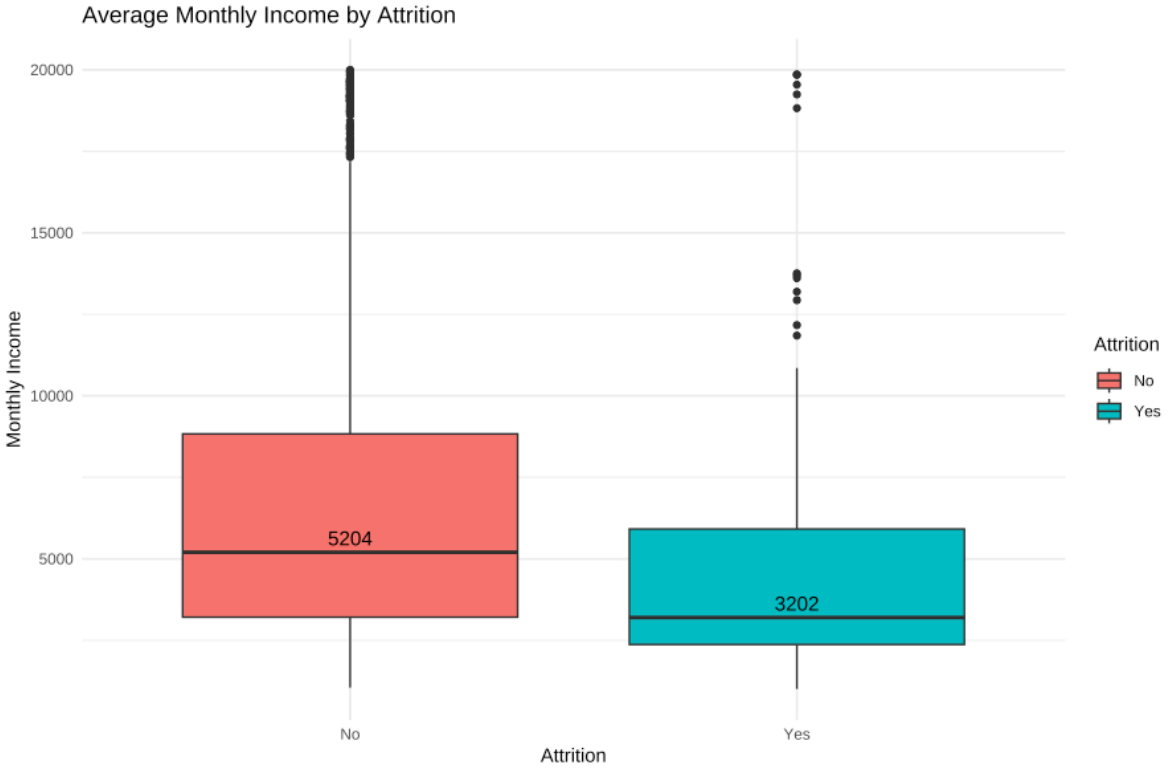

Figure 22: Monthly Income Analysis (Box Plot)

Salary has been and will always be a key factor, with churned employees having a lower average ($4.787.09) and median ($3,202) salary compared to employees staying with their respective organisations. However, the difference between the two groups is approximately

$2,000 in both measurements. This is further supported by the range of the box plot when comparing between the two groups, where the churned employees has a much lower salary cap. This indicates that salary discrepancy and bandwidth adjustment may be required.

Top 2nd - Overtime

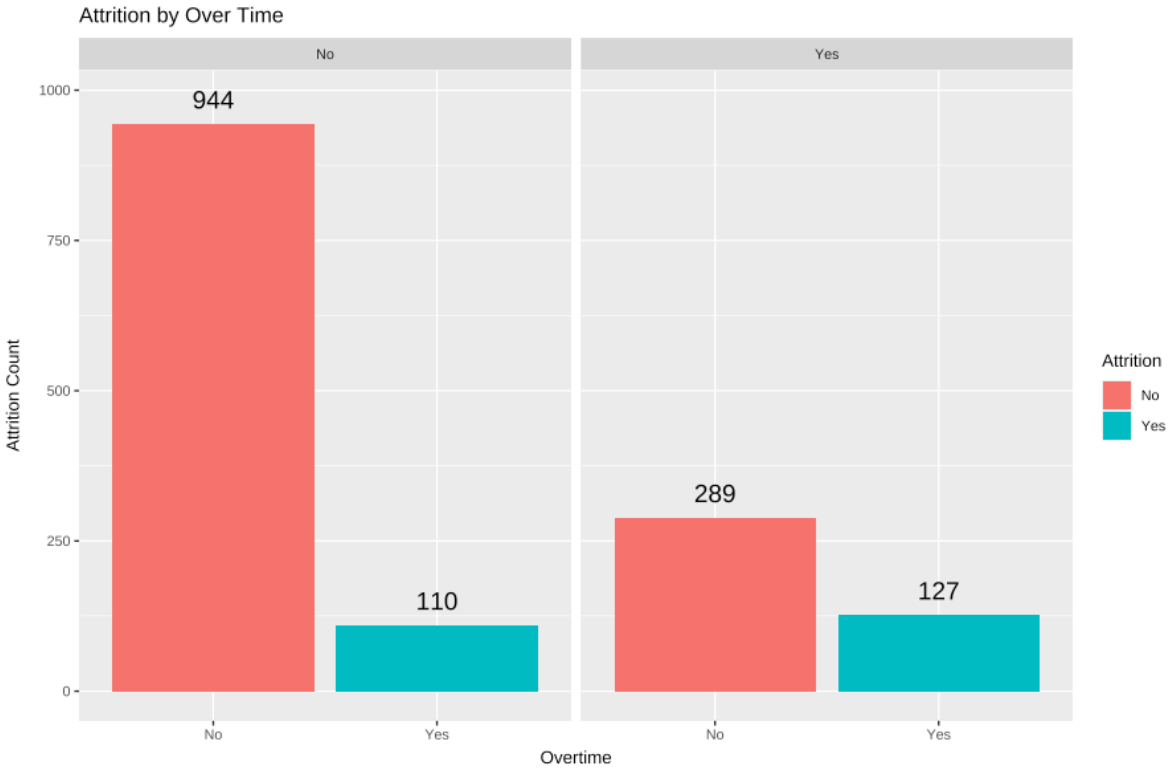

Figure 22: Attrition by Overtime Chart

From Figure 22, churned employees are noted to experience overtime or are overworked significantly more than those who stayed, with the overtime rate of retained employees at 23.44% as compared to churned employees at 53.59%. This could reveal a severe issue where a group of employees might constantly be experiencing crunch time, leading to burnout and ultimately, resignation. A further investigation of workload distribution and process review might be in order.

Top 3rd - Age

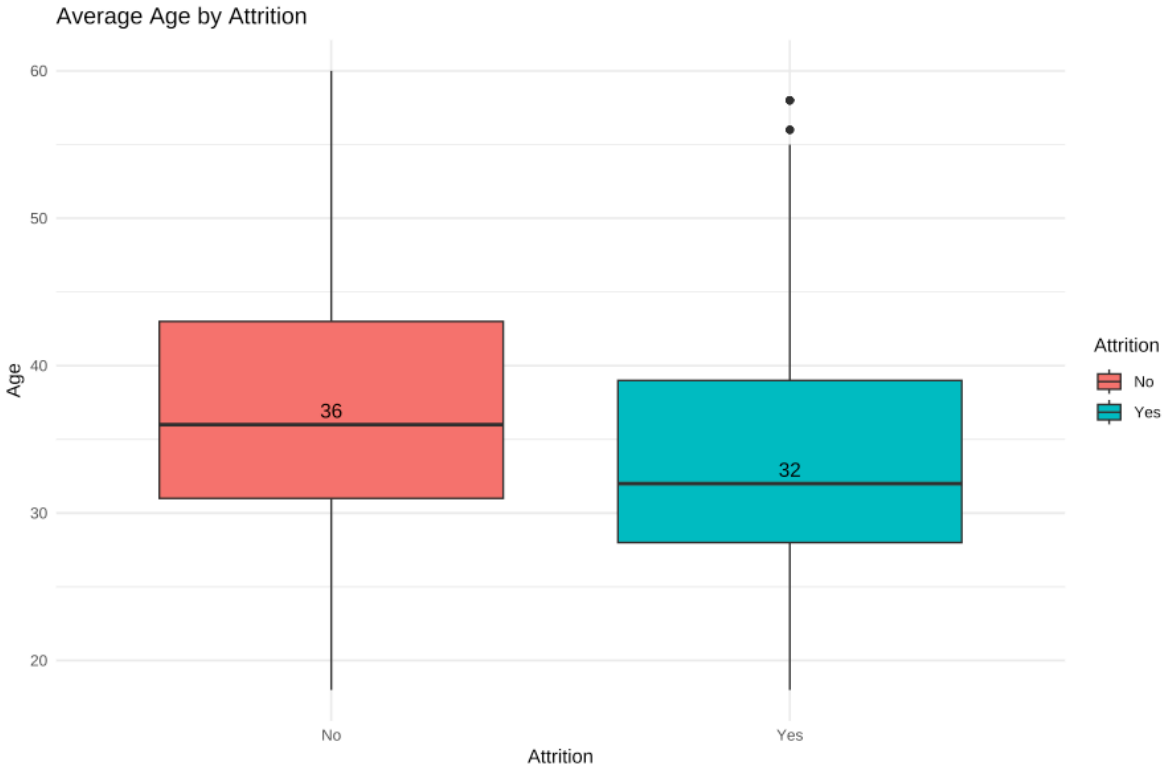

Figure 23: Age by Attrition (Box Plot)

The third most important factor for employee attrition is age. Employees who resigned tend to be on the younger end of the scale. This could be due to fresh graduates and younger employees leaving for other opportunities. Similarly, a lower maximum range (oldest) is significantly lower than the oldest employee who stayed in the role.

Top 4th - Total Working Experience

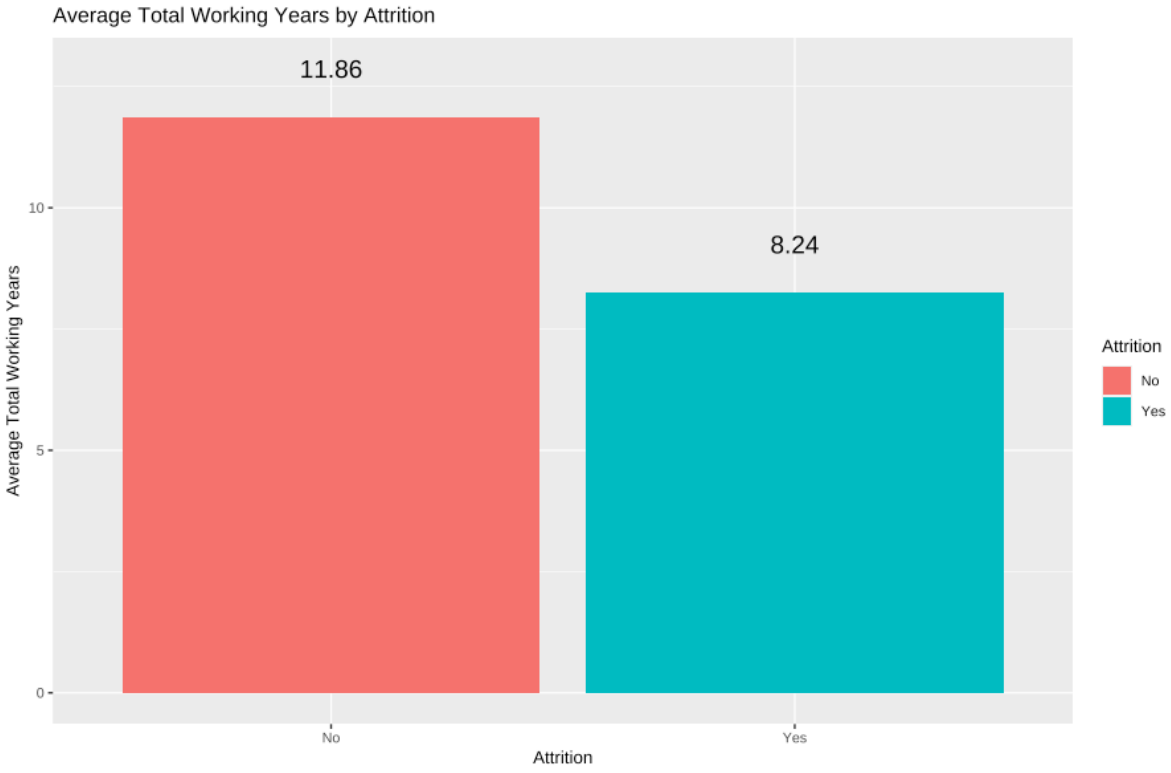

Figure 24: Total Working Years (Bar Chart)

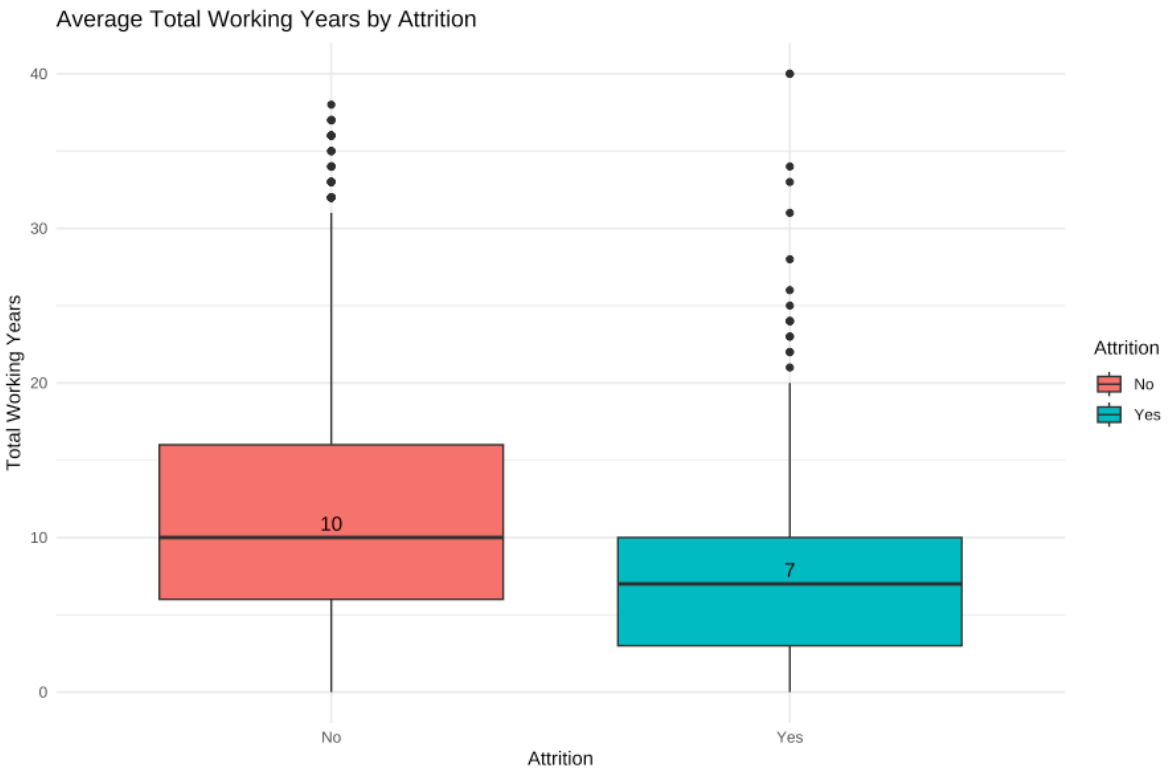

Figure 25: Total Working Years (Box Plot)

The Total Working Years of the employee is another crucial trait of employee attrition. This is similar to the findings in Figure 6 of data exploratory as experience would be proportionate to

the number of companies worked. Employees who resigned have an average of 8 and a median of 7 compared to their counterparts of 11.86 and 10, respectively. These lower values could be explained by employees that work longer being promoted to the management level where they would be inclined to stay. Therefore, those who remained in the organisations would have more years of experience.

Top 5th - Daily Rate

Figure 21: Daily Income Analysis (Box Plot)

Lastly, daily rate which is another financial indicator appears as a repeating key factor. Resignees tend to have a lower compensation with an average of $750 and median of $699 as compared to employees who remained with the organisation similar to monthly income.

However, the differentiator is that daily rate focuses on “Efficient Rate” which is subjective as compared to monthly income which is the total “Effective Rate” which is objective. The raise of daily rate factor could be attributed to the change in priorities of employees during the COVID-19 pandemic where more importance were being placed on personal wellbeing. As a result. the workforce have shifted towards the mentality of “earning more while working less”.

6.0 Conclusion

The Random Forest algorithm has successfully identified the top factors for employee churn, which could be helpful to organisations. These findings would assist the users in retaining key valuable employees and reduce employee churn and hiring costs. Apart from the financial impact, the retention of employees would reduce the loss of human knowledge capital often associated with an organisation's service quality, bringing about intangible benefits to the organisation.

In conclusion, both Logistics Regression and Random Forest can effectively identify traits and predict binary outcomes. However, Random Forest is the superior option for larger and more complex datasets because it can handle non-linear relationships and large numbers of input variables more effectively while reducing the risk of overfitting. Logistic Regression is still suitable for more straightforward problems with a small number of input variables and linear relationships between them. However, the algorithm of choice would ultimately boil down to each circumstance's unique requirement and desired outcome.

References

Abbasi, S. M., & Hollman, K. W. (2000). Turnover: The Real Bottom Line. *Public Personnel Management*, *29*(3), 333–342. https://doi.org/10.1177/009102600002900303

Booth, A. L., & Zoega, G. (1999). Do quits cause under-training? *Oxford Economic Papers-New Series*, *51*(2), 374–386. https://doi.org/10.1093/oep/51.2.374

IBM. (n.d.). *What is Logistic regression?* https://www.ibm.com/sg-en/topics/logistic-regression

Logistic Regression in Machine Learning. (n.d.). JavaTpoint. https://www.javatpoint.com/logistic-regression-in-machine-learning

Montaudon-Tomas, C. M., Amsler, A., Montaudon-Tomas, C. M., & Malcón-Cervera, C. (2022). Beyond the Great Resignation: Additional Notions. *The International Trade Journal*, *37*(1), 135–142. https://doi.org/10.1080/08853908.2022.2147107

Yiu, T. (2019, June 12). *Understanding Random Forest*. Towards Data Science. https://towardsdatascience.com/understanding-random-forest-58381e0602d2

Appendix A - Meta Data

1. Age: The age of the employee. (Numerical)
2. Attrition: Whether or not the employee has left the organization. (Categorical)
3. BusinessTravel: The frequency of business travel for the employee. (Categorical)
4. DailyRate: The daily rate of pay for the employee. (Numerical)
5. Department: The department the employee works in. (Categorical)
6. DistanceFromHome: The distance from home in miles for the employee. (Numerical)
7. Education: The level of education achieved by the employee. (Categorical)
8. EducationField: The field of study for the employee's education. (Categorical)
9. EmployeeCount: The total number of employees in the organization. (Numerical)
10. EmployeeNumber: A unique identifier for each employee profile. (Numerical)
11. EnvironmentSatisfaction: The employee's satisfaction with their work environment. (Categorical)
12. Gender: The gender of the employee. (Categorical)
13. HourlyRate: The hourly rate of pay for the employee. (Numerical)
14. JobInvolvement: The level of involvement required for the employee's job. (Categorical)
15. JobLevel: The job level of the employee. (Categorical)
16. JobRole: The role of the employee in the organization. (Categorical)
17. JobSatisfaction: The employee's satisfaction with their job. (Categorical)
18. MaritalStatus: The marital status of the employee. (Categorical)
19. MonthlyIncome: The monthly income of the employee. (Numerical)
20. MonthlyRate: The monthly rate of pay for the employee. (Numerical)
21. NumCompaniesWorked: The number of companies the employee has worked for. (Numerical)
22. Over18: Whether or not the employee is over 18. (Categorical)
23. OverTime: Whether or not the employee works overtime. (Categorical)
24. PercentSalaryHike: The percentage of salary hike for the employee. (Numerical)
25. PerformanceRating: The performance rating of the employee. (Categorical)
26. RelationshipSatisfaction: The employee's satisfaction with their relationships. (Categorical)
27. StandardHours: The standard hours of work for the employee. (Numerical)
28. StockOptionLevel: The stock option level of the employee. (Numerical)
29. TotalWorkingYears: The total number of years the employee has worked. (Numerical)
30. TrainingTimesLastYear: The number of times the employee was taken for training in the last year. (Numerical)
31. WorkLifeBalance: The employee's perception of their work-life balance. (Categorical)
32. YearsAtCompany: The number of years the employee has been with the company. (Numerical)
33. YearsInCurrentRole: The number of years the employee has been in their current role. (Numerical)
34. YearsSinceLastPromotion: The number of years since the employee's last promotion. (Numerical)
35. YearsWithCurrManager: The number of years the employee has been with their current manager. (Numerical)

Appendix B - User Guide

This is a guide on how to run our code (`ICT515 R Script.R`) file in RStudio.

Prerequisites
Before running the R file, you will need to have the following:

- RStudio installed on your computer
- The `ICT515 R Script.R` file saved on your computer

Instructions

1. Start by opening RStudio on your computer.
2. Ensure that the working directory is set to the folder where the `ICT515 R Script.R` file is located. To set the working directory, you can either use the "Set Working Directory" option in the "Session" menu or type setwd("path_to_folder") in the console, replacing "path_to_folder" with the actual path to the folder containing the file. Here are the example on how to set the working directory:

Set Working Directory
`# Change directory before run` `setwd("/Users/branata.kurniawan/Documents/personal/RStudio")`

3. Open the `ICT515 R Script.R` file by going to `File` > `Open File` in the top menu bar, or by clicking the "Open File" button on the "Files" tab in the bottom right pane of the RStudio window.
4. Install the packages that we are using on our code for the first time listed below. Or alternatively, you can uncomment the code line 8 to 18 by using `ctrl + shift + c` on windows or `cmd + shift + c` and click run afterward to install the packages.

Install packages

`install.packages("tidyverse")`
`install.packages("haven")`
`install.packages("readxl")`
`install.packages("readr")`
`install.packages("gridExtra")`
`install.packages("showtext")`
`install.packages("reshape2")`
`install.packages("caret")`
`install.packages("randomForest")`
`install.packages("rpart")`
`install.packages("rpart.plot")`

5. Next is to update the path on where the CSV file is located. The file will be in the same folder as the R script. The name of the CSV file is `HR_Analytics.csv`. Here are the example on how to change the file path:

Update the path to the data file

`# Read the data`
`data <-`
`read_csv("/Users/branata.kurniawan/Documents/personal/ICT515/HR_Analytics.c`

```
sv")
```

6. Lastly, to run the code in the `ICT515 R Script.R` file, you can either click the "Source" button located above the editor pane, or you can use the keyboard shortcut `ctrl + shift + s` on Windows or `cmd + shift + s` on Mac. This will run all of the R code in the file.

More Information

With these user guides, you should be able to run the `ICT515 R Script.R` file in RStudio. If you have any issues or questions, please drop us an email on branata007@gmail.com or ericngzh96@gmail.com.