IBM HR ANALYSIS

TEAM BARBARIANS

BU.330.780.T1.SP21
Data Science and
Business Intelligence

Minghao Du, Mengying Zhao, Wenlu Chen, Haitong Ni, Haoyu Wang

Cracking Attrition: A Solution Finding Through IBM HR Analysis

Minghao Du, Mengying Zhao, Wenlu Chen, Haitong Ni, Haoyu Wang

Johns Hopkins University

BU.330.780.T1.SP21: Data Science and Business Intelligence

Dr. Changmi Jung

May 17, 2021

Table of Contents

Business Understanding 4

Data Understanding & Preparation 5

Visualization 6

Modeling 7

Model 1: Decision Tree Model 7

Model 2: Random Forest Model 8

Model 3: SVM 9

Evaluation 11

Deployment 12

Reference 15

Appendix 16

Business Understanding

High attrition can cause many huge, serious problems within an organization. In addition to the unavoidable cost of training and other expenses of new employees, the company must also bear a lot of risks, for example, the sudden departure of employees will result in the interrupted of executing tasks, and the company's operating efficiency will be greatly reduced. All of these will have a negative impact on the company's future development.

In most cases, continuous evaluation of employees' work can yield a lot of different data and variables. In this case, our target variable is attrition. Employee resignation is composed of many factors. In order to reduce the number of Employee resignation, thereby reducing the risks and costs caused by it, and improving the company's operational efficiency, we need to find out the factors that are highly relevant to employee attrition. Through analyze many data characteristics, such as "Education field", "Distance from home", "Job role", "Job satisfaction", "Work-life balance", and so forth, the goal could be obtained. This can help the company better planning and developing solutions to reduce the possibility of employee attrition. For example, if resignation is related to low salary, the company could actively adjust salaries to retain good-performing employees.

Since the important factors affecting attrition have not been defined at the beginning, we do not predict and formulate optimization plans to reduce employee attrition in the initial stage. Instead, we with obtain effective information through data mining, and finally propose recommendations and plans combined with its results.

Data Understanding & Preparation

The data used to address our business problem was retrieved from Kaggle: IBM HR Analytics Employee Attrition & Performance.
https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset
In the source data, the data fields are demonstrated in **Exhibit 1**.

By looking at the problem we need to solve and observing the data set features, this is a classification problem. The training data set includes 35 variables (columns) and 1470 rows. The variables “Attrition” is our target variable, it means that whether the employee resigned or not. We first focus on sort out the useful variables. After the initial observation, the variables “Employee Count”, “Standard Hours”, “Over 18” have the same answer in the data set. For example, the results in whether employees are over 18 in the company are all “Yes”. Therefore, these three variables are meaningless, we need to drop them at the very beginning of the data preparation.

By observing the characteristics of the data, some variables cannot be quantitatively processed. In order to quantify those variables, to better analyze the data, we created some dummy variables. Split the variables of a certain category, for instance, setting "Gender" to two dummy variables "Gender_Male" and "Gender_Female". After applying the dummy variables and cleaning the data (such as removing useless characters, See **Exhibit 2** for the code), we have prepared the initial data with 53 columns that can be used for initial prediction and put into the model for further analysis.

Visualization

To better understand the data set, we plot several histograms for the raw data. Our target variable contains more than 1200 “No” and around 220 “Yes”, which means around 15% of our data set was Attrition **(Exhibit 3).**

In **Exhibit 4**, the histogram between Attrition and Age demonstrates that the most of employees are in their 25-35; 25-35 years-old has the largest number of Attrition as well as Not Attrition.

In **Exhibit 5**, we plot the histogram for Monthly Income and Attrition number. The $1000-$2000 level contains the highest number of Attrition, while almost no Attrition since the Monthly Income reached $15000.

In **Exhibit 6** the bar chart between Over Time and Attrition displays the relatively equal number of Over Time in both columns. The Over Time is higher in “Yes” column than “No”, which indicates that the extra working time might increase the Attrition.

In **Exhibit 7**, the bar chart displays that the Research & Development has the greatest number of employees, Sales has the second greatest number of employees. Also, Research & Development contains the greatest number of Attrition, Sales has the highest has the second greatest number of Attrition.

In **Exhibit 8**, the number of male employees is larger than female employees, while the number of Attrition for male employees is more than female’s Attrition.

In **Exhibit 9**, the scatterplot between the Monthly Income and Age illustrates that employee who has lower monthly income and younger age are more likely to be Attrition. On the other hand, we do have more dense data in the lower monthly income and younger age, and sparse data distribution higher monthly income and older age, which might bring influence on the prediction.

In **Exhibit 10**, the scatterplot between the Total Working Years and Years at Company reveals that employee who has less Total Working Years and less Years at Company are more likely to be Attrition. However, we do have more dense data in less Total Working Years and less Years at Company, and more sparse data distribution higher Total Working Years and higher Years at Company, which might also impact the prediction.

Modeling

Model 1: Decision Tree Model

Ross proposed the concept of decision trees in the 19th century. Although decision trees are relatively simple models, they are highly interpretable. The decision tree represents a certain mapping between object attributes and object values. It includes nodes and roots. The nodes in the decision tree represent samples, internal nodes represent elements or attributes, and leaf nodes represent classes.

The main goal of our project is to predict which types of employees are more likely to decide attrition and which factors will most affect employees' decision to leave. We decided to build the first model-decision tree to explore the interaction between factors. First, we randomly use half of the data as the training dataset and the other half as the testing dataset. According to the performance of the decision tree we built on the training dataset, we can conclude that monthly income, years with current manager, work life balance, job satisfaction, over time, stock option level environment satisfaction and job role have a greater impact on employee attrition. Specifically, employees with monthly income less than 2805, over time, environment satisfaction less than 3 and work life balance larger than 2 are more likely to make attrition decisions. The accuracy rate for the first decision tree model is 0.8125 **(Exhibit 11)**. After that, we try to limit the characteristics of the decision tree and change some criteria to find the best decision tree to predict our target variable. In this decision tree, we found that monthly income and over time can best represent to predict attrition or not. In detail, employees with monthly income less than 2805

and over time are more likely to choose attrition **(Exhibit 12)**. Based on the result, we know that our best decision tree model's accuracy rate is 0.8478, which is higher than our first model.

Model 2: Random Forest Model

In this dataset, the target is to predict the attrition result of employees under different situations, including hour rate, environment, age, job satisfaction and so many other factors. Random Forest model is effective and flexible by allowing each individual tree to randomly sample from the dataset with replacement, resulting in different trees.

Each individual tree in the random forest spits out a class prediction and the class with the most votes become our model's prediction. In this investigation, we have 1470 uncorrelated samples, and we created a validation set by using 75% of samples as our training data and the rest of it as our testing data, which is for the model not overfitting. Firstly, we tested the hyper parameter "mtry" as the **(Exhibit 13)** shows.

When training data is ready, the (out-of-bag) data is for estimating the classification error as trees, which is also used to get estimates of variable importance. The OOB estimate of error rate is 14.16%, and the result is unbiased **(Exhibit 14)**.

The feature importance's in a random forest can help us figure out what predictor variables the random forest considers most important. According to our test, there are 11 relevant variables that are critical to our target, including: OverTime_No, MonthlyIncome, Age, JobRole, Marital_Status, YearsSinceLastPromotion, TotalWorkingaYears, JobRole, StockOptionLevel, JobSatisfaction, RelationshipSatisfaction **(Exhibit 15)**.

Then, we used Brute-force search for best mtry, and we have obtained that mtry equals to 2 through this method. In general, Brute-force search is useful as a baseline method when benchmarking other algorithms or metaheuristics. Since our dataset has limited samples, we can use this method efficiently to get results.

According to our model, another diagnostic measure of the model we can take is to plot the confusion matrix for the testing predictions. We can see in the model that, based on predictions, how many got correct in the top left and bottom right corners and how many were missed in the lower left and upper right. In summary, the accuracy rate of our random forest model is 0.8777, which outperforms the single decision tree **(Exhibit 16)**.

Model 3: SVM

The Support Vector Machine (SVM) model is a supervised machine learning model that can classify two (or more) factors target variable depending on the choice of kernels. In our case, the processed data presents 52 independent variables and one target variable (Attrition); thus, while fitting to the training data, the SVM model will generate the best hyperplane (with highest margin between vectors in different groups) in a 53-dimensional space.

For the first attempt we passed all training data to a SVM model with a radial kernel. We then apply the model to the preserved testing data to generate a set of predicted Attrition values. This predicted attrition value is used to compare with the actual attrition value in the testing data to calculate the accuracy value. Detail confusion matrix and statistics shown below:

True Positive : 15

True Negative : 317

False Positive : 1

False Negative : 35

Accuracy : 0.9022

95% CI : (0.8671, 0.9305)

Sensitivity : 0.9006

Specificity : 0.9375

Pos Pred Value : 0.9969

Neg Pred Value : 0.3000

Prevalence : 0.9565

Detection Rate : 0.8614

Detection Prevalence : 0.8641

Balanced Accuracy : 0.9190

Given the radial kernel is sensitive to cost and gamma (hyperparameter), the model might benefit from fine tuning with high resolution hyperparameters. Hence, for cost, we prepared a sequence ranging from 1e-5 to 1e4 with 10x increment, and for gamma, we prepared a sequence ranging from 1e-4 to 10 with 5x increment. Applying the combination of these two hyperparameters, the best SVM model shall select from 180 candidate models. As a result, the outputted best SVM model is tuned with cost of 100 and gamma of 5e-4.

Applying the tuned model to the preserved testing data set, a set of predicted attrition value is generated. These predicted values are then checked against the original ones to calculate the accuracy of the model. Detail confusion matrix and statistics shown below:

True Positive: 24

True Negative: 312

False Positive: 6

False Negative: 26

Accuracy: 0.913

95% CI: (0.8795, 0.9398)

Sensitivity: 0.9231

Specificity: 0.8000

Pos Pred Value: 0.9811

Neg Pred Value: 0.4800

Prevalence: 0.9185

Detection Rate: 0.8478

Detection Prevalence: 0.8641

Balanced Accuracy: 0.8615

As the result shown, after tuning the model a slight increase of accuracy is emerged. On the other hand, while comparing the balanced accuracy of both models, the improvement is significant. As the matter of utilizing the model, both are equally good.

Evaluation

At this point, all three models had built, tuned, and tested against the preserved testing data set. Below is the detail confusion matrix and statistics for all three models:

	Decision Tree	Random Forest	SVM
True Positive	8	5	23
True Negative	304	318	312
False Positive	14	0	6
False Negative	42	45	27
Accuracy	0.8478	0.8777	0.9103
Sensitivity	0.9560	1.0000	0.9811
Specificity	0.1600	0.1000	0.4600
Pos Pred Value	0.8786	0.8760	0.9204
Neg Pred Value	0.3636	1.0000	0.7931
Prevalence	0.8641	0.8641	0.8641
Detection Rate	0.8261	0.8641	0.8478
Detection Prevalence	0.9402	0.9864	0.9212
Balanced Accuracy	0.5580	0.5500	0.7206

As shown in the above table, we can say that the SVM model has the highest accuracy rate, which arrives at 91%. The random forest model has the second highest accuracy rate, and the decision tree model has the lowest accuracy rate. AUC is also an important fact to show how our model performs in the testing dataset. Based on the **Exhibit 17,** it shows AUC values among three different models. Black line represents SVM model performance and green line represents random forest model and blue line represents decision tree model performance. We can clearly say that the random forest and SVM model's performance are much better than the decision tree model. Compared with the random forest model, although at some point, the random forest model has higher sensitivity, overall, the SVM model has higher AUC value (0.835). Therefore, our group thinks the SVM model is the best model we created.

From a practical standpoint, both SVM and random forest are equally good in terms of predicting employee attrition. However, when it comes to formulating plans to decrease attrition rate, random forest provides more relevant information like variable importance and prediction visualizations. Hence, depending on the application, both model offers different perks for different aspect of the problem that the firm is facing.

Deployment

According to the above evaluation, Random Forest and SVM perform better in these three models, so the next we will conduct further analysis based on Random Forest.

Based on the results of the random forest above, there are 11 variables in total that are significant relative to the attrition. We choose the first four variables since they might be the most important to decide whether employee resign or not. They are “Overtime”, “Monthly income”, “Age” and “Total working years”.

Among all the four factors, the most significant variable that led to employee attrition is overtime. By observing the chart, it can be found that the higher the overtime, the greater the possibility of employees leaving **(See Exhibit 18)**. And through analysis, under the premise of high overtime rate, employees whose monthly salary is less than approximately 2500 dollars, the age is less than 30, and the total working year is less than 5 years are more likely to leave. But in general, high overtime rates will increase attrition **(See Exhibit 19, 20, 21)**. Therefore, in view of this feature, the company can consider reducing the number of overtime work. For overtime that cannot be avoided, the company can conduct further investigations internally to summarize how employees can reduce negative emotions under the overtime circumstances, such as adding additional work allowances or benefits to overtime employees and so on. Using these to balance the extra overtime that employees pay. Through such incentive arrangements, it is expected that the negative impact of overtime can be minimized, thereby reducing the possibility of employees leaving due to it.

Through continued analysis, it can be found that when the monthly income is lower than 2500 dollars, no matter how long the employee's working experience, the probability of leaving the job is very high. Compared with older employees, young people who aged lower than 27 are relatively more likely to resign under low wages **(See Exhibit 22)**. According to this, the company may have to reconsider the issue of salary setting, such as raising the minimum salary, which needs to be in line with the current talent market conditions. In this way, employees will not leave their jobs because of salary, when they receive a salary that is equal to the value of their work.

The last two important factors that affect the company's departure are age and total working years. There is a positive correlation between these two factors, that is, the younger the age, the shorter total working years **(See Exhibit 23)**. Therefore, these two factors are combined as a reason for the impact of resignation. We found that employees who are younger and have a shorter total working years are more likely to quit the job. It can be presumed from this result that because young people have not worked for a long time in the company, and some may even be the new employees, they are very positive about the future development and have high expectations for the company. However, if employees have a certain distance between their expectations and reality of the company, they will feel a sense of gap and worry about their current work and job prospects. Therefore, companies should provide employees with more realistic work information, including remuneration, working environment, conditions, etc., to ensure that employees form appropriate expectations. Then the attrition might decrease after applying this suggestion.

However, there are still several problems with the above suggestions. For example, the collected past data might be biased. Resigned employees usually avoid telling the real reason for their resignation, as well as the dissatisfaction with the company. It does more harm than good to them. Therefore, the solutions we propose may not be able to neutralize the problems of which caused by the actual reason leads to resignation. At the same time, since the resignation is a very

subjective matter, it is usually not determined solely by these factors, so the company should keep tracking the resigned employees, in order to get a better prediction in the future.

And our model has many aspects that can be improved. For example, we didn't apply the horizontal comparison of each of our models, and the same model does not have a compare of different kernels, which will bring about different performance and a comparison with accuracy. In addition, the current highest accuracy rate of our model is 91%, which could be further optimized. At present, we have selected only three models, but there are many other models such as neural network that are not used, there might be better results in these other models.

In conclusion, coming from a problem-solving perspective, the project had followed appropriate data analysis procedure. Thus, regardless of the imperfection of the current model, it is sufficient, in terms of both accuracy and variable evaluation, to formulate an effective plan to address the rising business problem. On the other hand, taking a closer look at the current model and the solution of which it derived, the team thinks that it would nor be cost effective to further improve the model since the diminishing return on increase of accuracy will not provide better insight into those independent variables. Therefore, even though there might be rooms to better existing models, the firm might be more beneficial from executing the recommending solution.

Reference

GitHub & BitBucket HTML Preview. (n.d.). Retrieved May 17, 2021, from https://htmlpreview.github.io/?https://github.com/geneticsMiNIng/BlackBoxOpener/blob/master/randomForestExplainer/inst/doc/randomForestExplainer.html

Gupta, P. (2017, November 12). Decision Trees in Machine Learning. Retrieved May 17, 2021, from https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052

IBM HR Analytics Employee Attrition & Performance. (n.d.). Retrieved May 17, 2021, from https://kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset

Prabhakaran, S. (n.d.). Top 50 ggplot2 Visualizations - The Master List (With Full R Code). Retrieved May 17, 2021, from http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html

R: Plot a ROC curve with ggplot2. (n.d.). Retrieved May 17, 2021, from https://search.r-project.org/CRAN/refmans/pROC/html/ggroc.html

RPubs - K-Fold Cross Validation applied to SVM model in R. (2019, June 20). Retrieved May 17, 2021, from https://rpubs.com/markloessi/506713

RPubs - SVM modelling with parameter tuning and feature selection using Pima Indians Data. (2017, August 4). Retrieved May 17, 2021, from https://rpubs.com/Kushan/296706

Support Vector Machines Using svm() function. (n.d.). Retrieved May 17, 2021, from https://rstudio-pubs-static.s3.amazonaws.com/271792_96b51b7fa2af4b3f808d04f3f3051516.html

Yihui Xie, J. (2021, April 9). 3.1 HTML document | R Markdown: The Definitive Guide. Retrieved May 17, 2021, from https://bookdown.org/yihui/rmarkdown/

Appendix

Exhibit 1: Data Characteristics from Kaggle.com for IBM HR Analytics Employee Attrition & Performance.

Age – Employee age

Attrition – Whether the employee is still in position.

Business Travel -The frequency of business travel

Daily Rate - An internal measure in cost of employment per day (e.g., salary, insurance, logistics, overhead, etc.)

Department - Different departments in IBM

Distance From Home - The distance from employee's home to company

Education - The education level of the employees (1: Below College, 2: College, 3: Bachelor 4: Master 5: Doctor)

Education Field - The focus major of the employees

Employee Number - Employee ID

Environment Satisfaction - The level of satisfaction that employee have on their working environment. (1: 'Low'; 2: 'Medium'; 3: 'High'; 4: 'Very High')

Gender - Female or Male

Hourly Rate - An internal measure in cost of employment per hour (e.g., salary, insurance, logistics, overhead, etc.)

Job Involvement - a state of psychological identification with work—or the degree to which a job is central to a person's identity. (1: 'Low'; 2: 'Medium'; 3: 'High'; 4: 'Very High')

Job Level - set the responsibility level and expectations of roles.

Job Role - The different functions you fill within the organization.

Job Satisfaction - How satisfied an employee. (1: Low, 2: Medium, 3: High, 4: Very High)

Marital Status - Employee Marital Status

Monthly Income - Monthly Salary

Monthly Rate - An internal measure in cost of employment per month (e.g., salary, insurance, logistics, overhead, etc.)

Num Companies Worked - number of previous employers.

Over Time - length of extended working hours

Percent Salary Hike - The parentage of change in salary between 2 year (2017, 2018).

Performance Rating - Numerical Value - ERFORMANCE RATING (1: Low, 2: Medium, 3: High, 4: Very High)

Relationship Satisfaction - Numerical Value – How satisfied an employee about the company (1: Low, 2: Medium, 3: High, 4: Very High)

Standard Hours - How many hours

Stock Option Level - numerical value - stock options. The company stocks people own from this company.

Total Working Years - Total number of years at the company

Training Times Last Year - numerical value - hours spent training

Work Life Balance - numerical value - time spent between work and outside (1: bad; 2: good; 3: better; 4: best)

Years at Company - numerical value - total number of years at the company

Years in Current Role - years in current role

Years Since Last Promotion - numerical value - last promotion

Years with Current Manager - numerical value - years spent with current manager.

Exhibit 2

```r
tmp_Attrition ← raw_data$Attrition

data_droped_col ←
  subset(raw_data,
         select = -c(Attrition, EmployeeCount, StandardHours, Over18))

data_dummied ← dummy_cols(data_droped_col)

data_processed ←
  data_dummied[, sapply(data_dummied, class) ≠ "factor"]

colnames(data_processed) ← gsub(" ", "", colnames(data_processed))
colnames(data_processed) ← gsub("-", "", colnames(data_processed))
colnames(data_processed) ← gsub("&", "", colnames(data_processed))
colnames(data_processed) ← gsub("`", "", colnames(data_processed))

data_processed$Attrition ← tmp_Attrition

glimpse(data_processed)
```

Exhibit 3

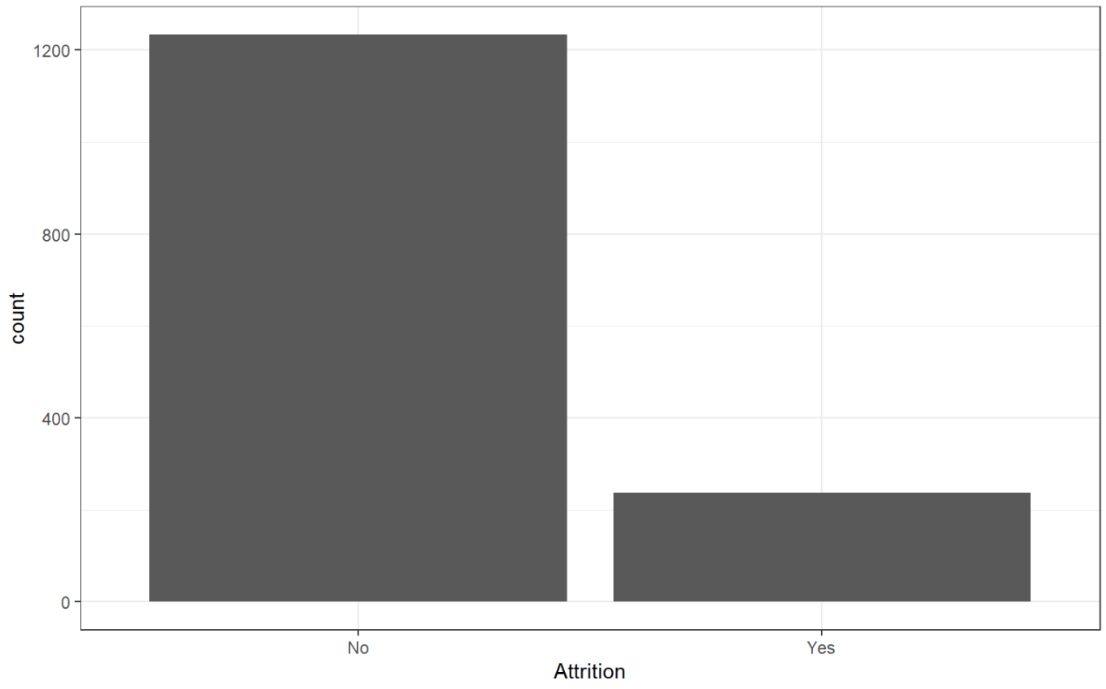

Exhibit 4

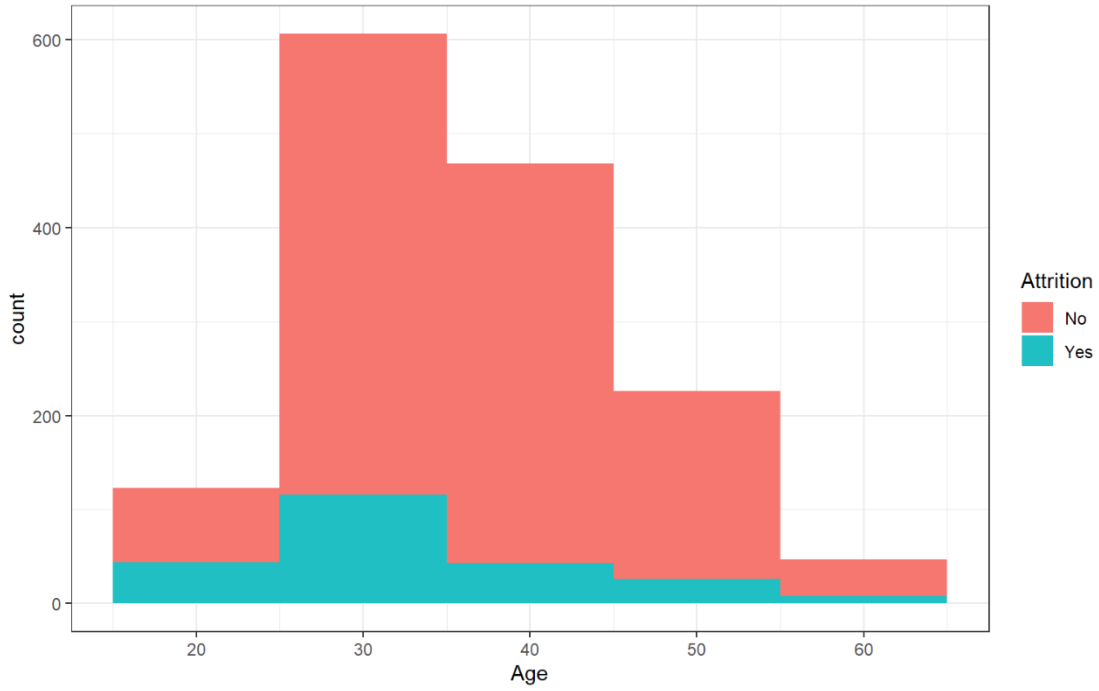

Exhibit 5

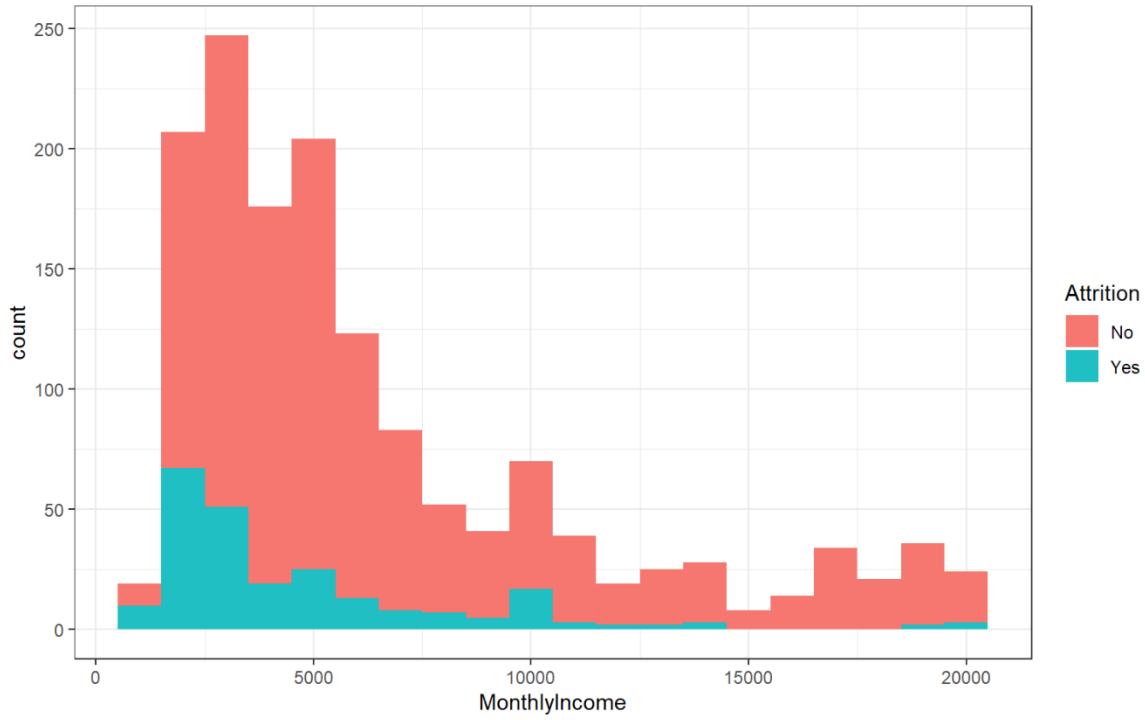

Exhibit 6

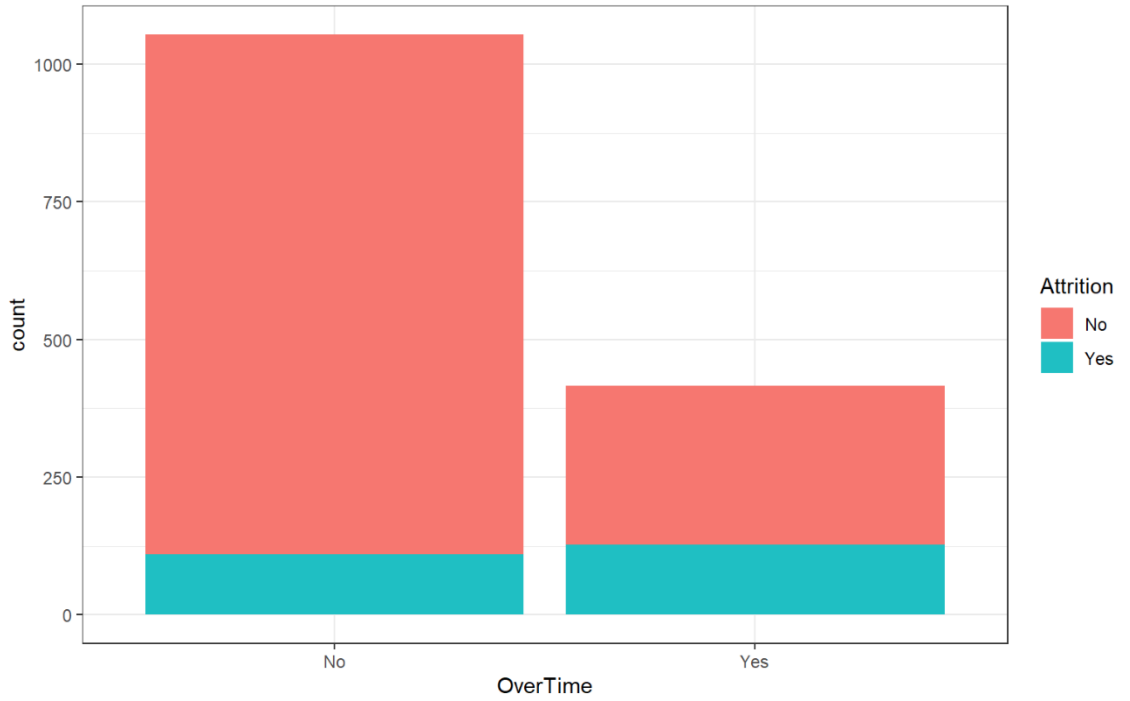

Exhibit 7

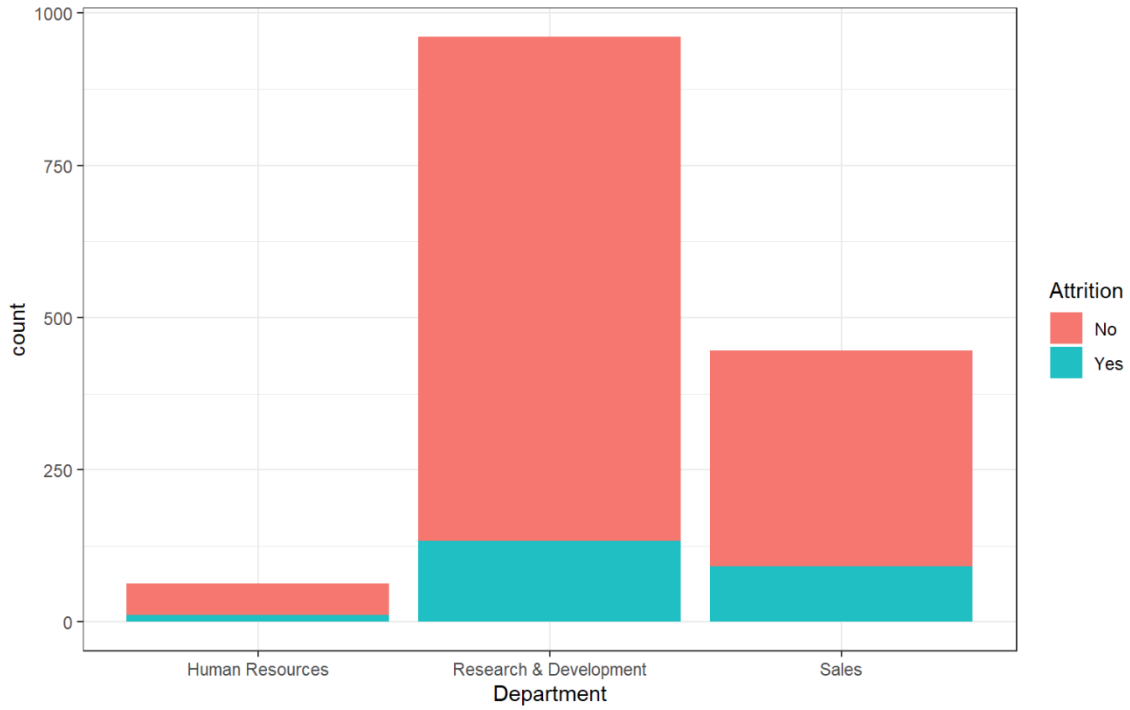

Exhibit 8

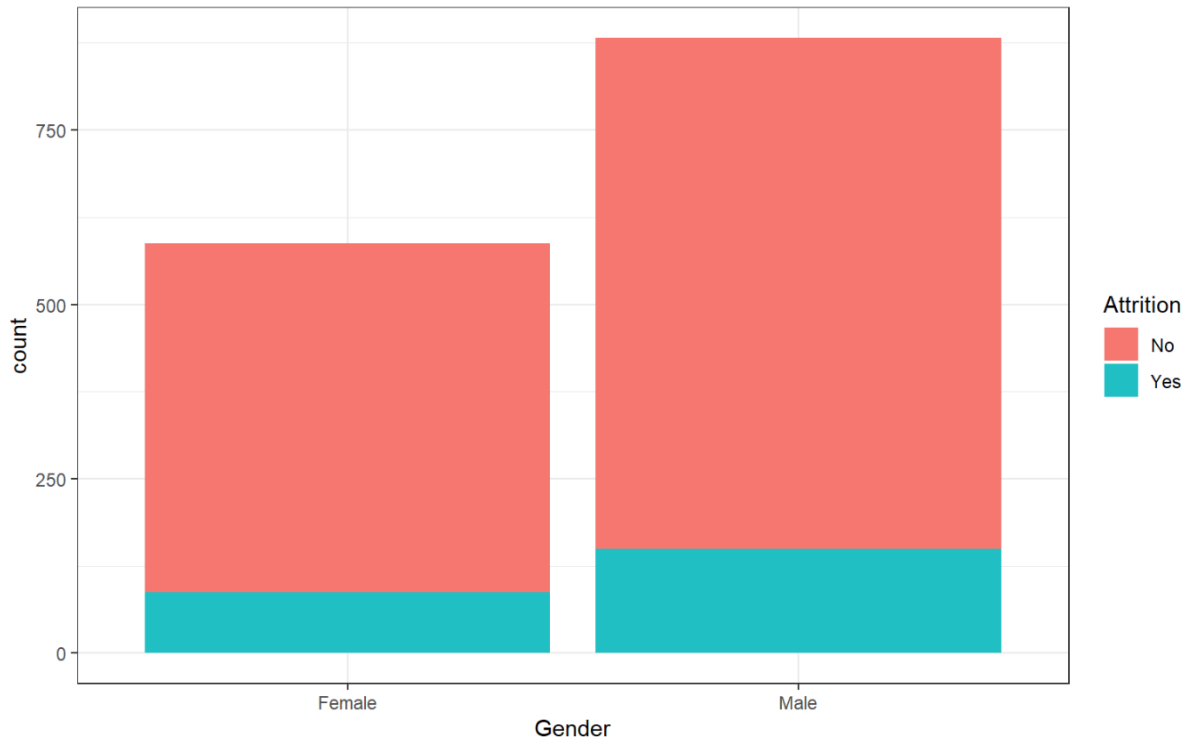

Exhibit 9

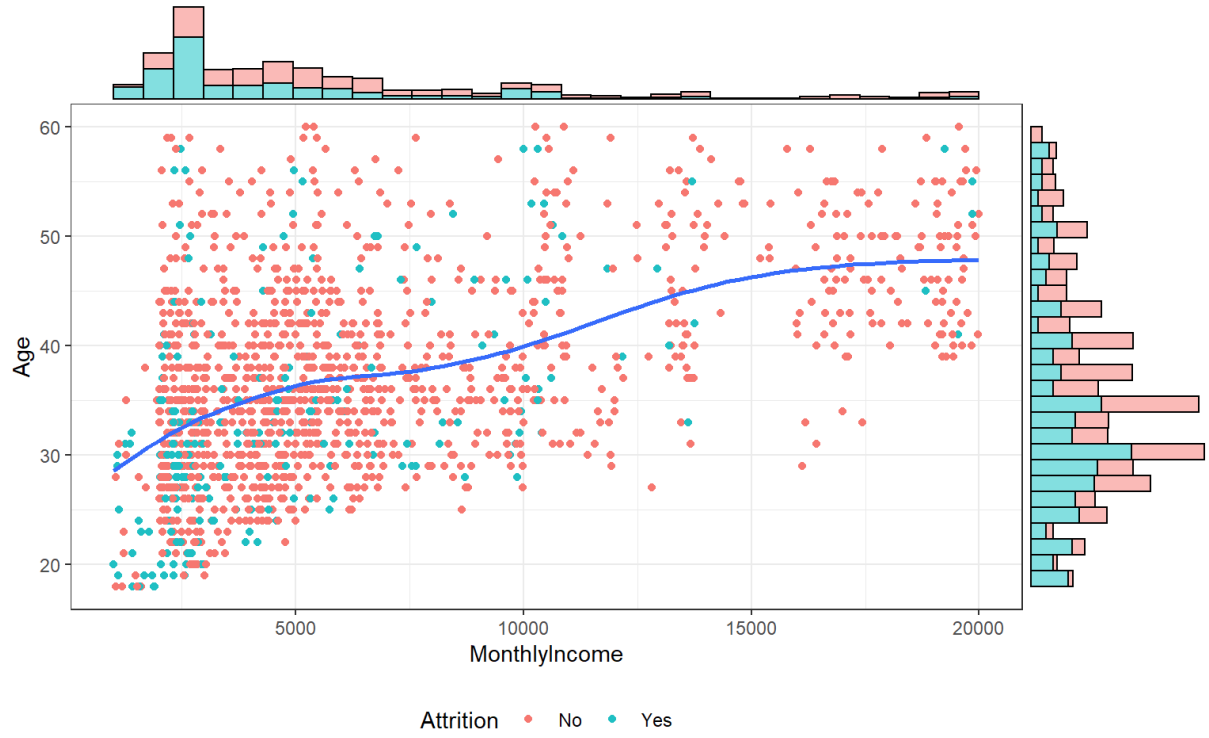

Exhibit 10

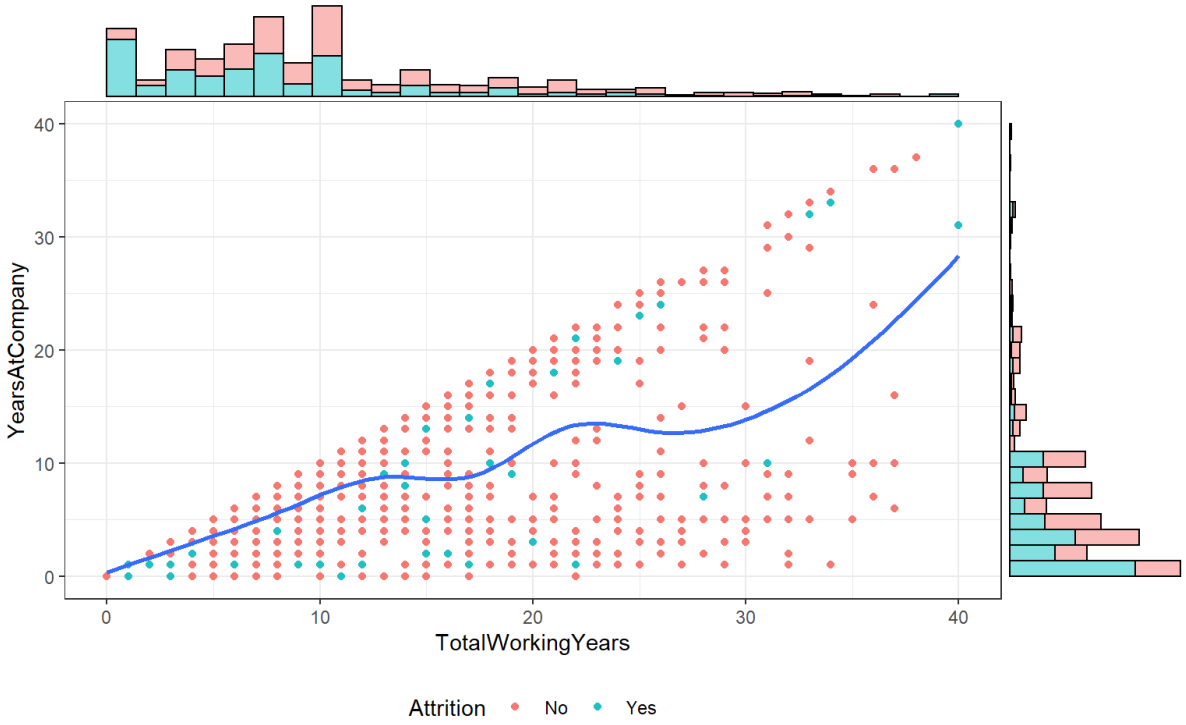

Exhibit 11

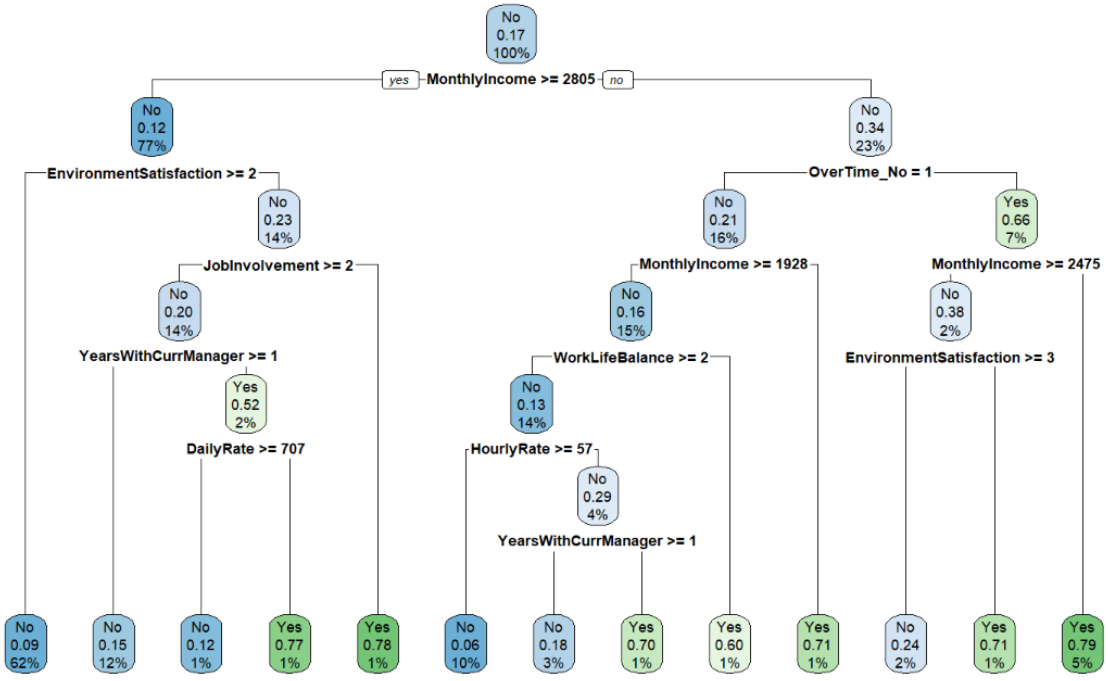

Exhibit 12

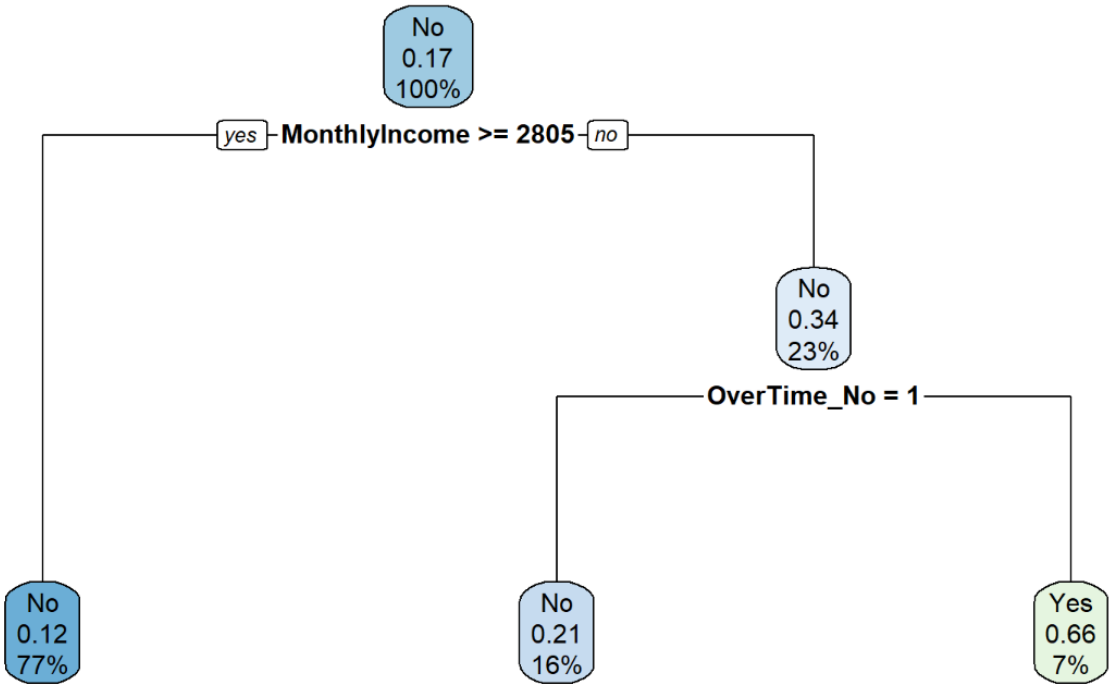

Exhibit 13

```
data.forest ← randomForest(
  Attrition ~ .,
  data = data_train,
  ntree = treeCount,
  mtry = ncol(data_processed) - 1,
  importance = T,
  localImp = T
)
data.forest
```

Exhibit 14

```
##
## Call:
##  randomForest(formula = Attrition ~ ., data = data_train, ntree = treeCount,
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 52
##
##         OOB estimate of  error rate: 14.16%
## Confusion matrix:
##      No Yes class.error
## No  894  21  0.02295082
## Yes 135  52  0.72192513
```

Exhibit 15

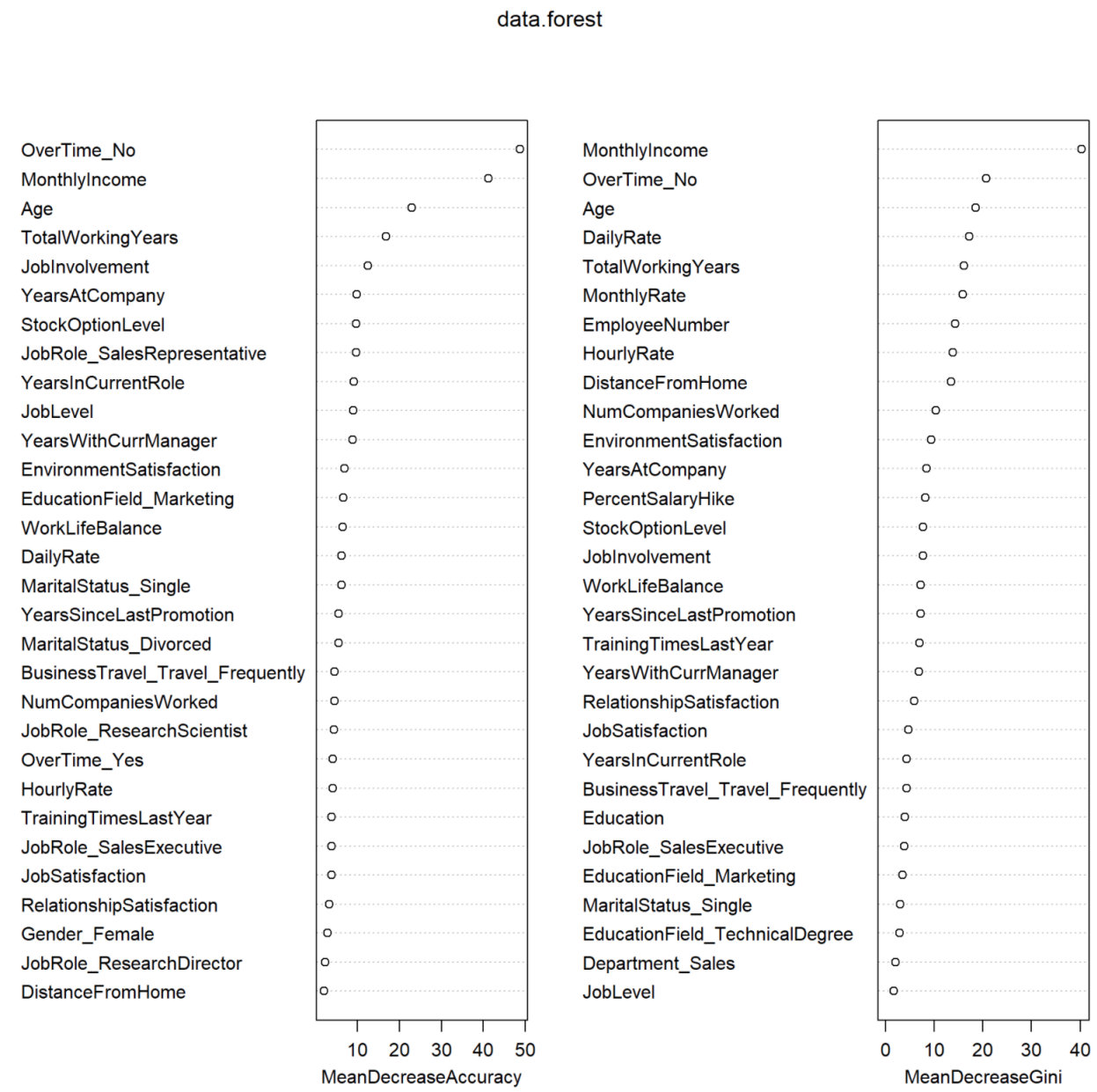
data.forest
OverTime_No
MonthlyIncome
Age
TotalWorkingYears
JobInvolvement
YearsAtCompany
StockOptionLevel
JobRole_SalesRepresentative
YearsInCurrentRole
JobLevel
YearsWithCurrManager
EnvironmentSatisfaction
EducationField_Marketing
WorkLifeBalance
DailyRate
MaritalStatus_Single
YearsSinceLastPromotion
MaritalStatus_Divorced
BusinessTravel_Travel_Frequently
NumCompaniesWorked
JobRole_ResearchScientist
OverTime_Yes
HourlyRate
TrainingTimesLastYear
JobRole_SalesExecutive
JobSatisfaction
RelationshipSatisfaction
Gender_Female
JobRole_ResearchDirector
DistanceFromHome
10 20 30 40 50
MeanDecreaseAccuracy
MonthlyIncome
OverTime_No
Age
DailyRate
TotalWorkingYears
MonthlyRate
EmployeeNumber
HourlyRate
DistanceFromHome
NumCompaniesWorked
EnvironmentSatisfaction
YearsAtCompany
PercentSalaryHike
StockOptionLevel
JobInvolvement
WorkLifeBalance
YearsSinceLastPromotion
TrainingTimesLastYear
YearsWithCurrManager
RelationshipSatisfaction
JobSatisfaction
YearsInCurrentRole
BusinessTravel_Travel_Frequently
Education
JobRole_SalesExecutive
EducationField_Marketing
MaritalStatus_Single
EducationField_TechnicalDegree
Department_Sales
JobLevel
0 10 20 30 40
MeanDecreaseGini

Exhibit 16

```
Confusion Matrix and Statistics

          actual
predicted  No Yes
      No  318  45
      Yes   0   5

               Accuracy : 0.8777
                 95% CI : (0.8398, 0.9094)
    No Information Rate : 0.8641
    P-Value [Acc > NIR] : 0.2499

                  Kappa : 0.1611

 Mcnemar's Test P-Value : 5.412e-11

            Sensitivity : 1.0000
            Specificity : 0.1000
         Pos Pred Value : 0.8760
         Neg Pred Value : 1.0000
             Prevalence : 0.8641
         Detection Rate : 0.8641
   Detection Prevalence : 0.9864
      Balanced Accuracy : 0.5500

       'Positive' Class : No
```

Exhibit 17

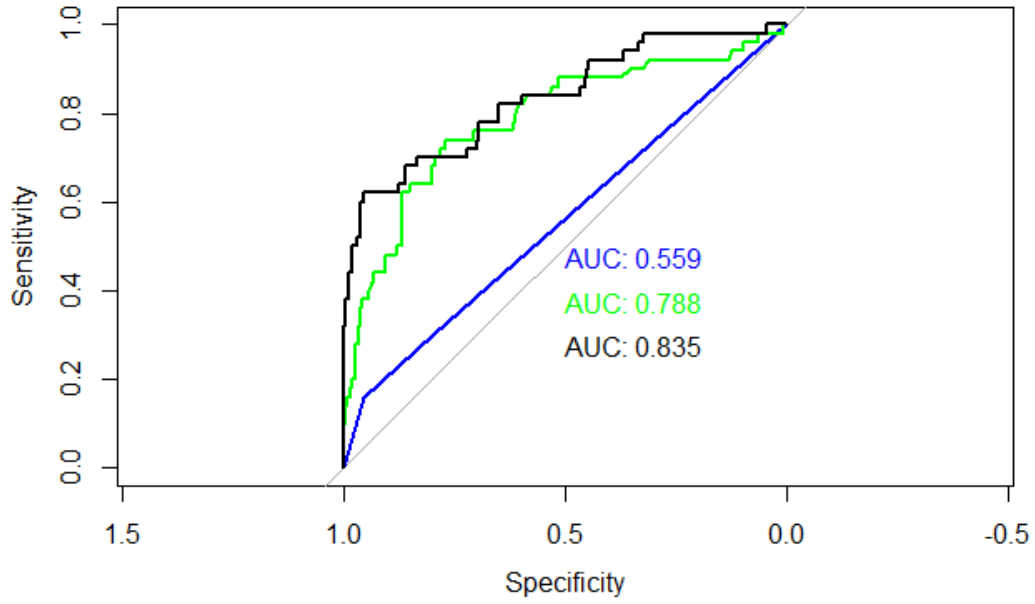

Exhibit 18

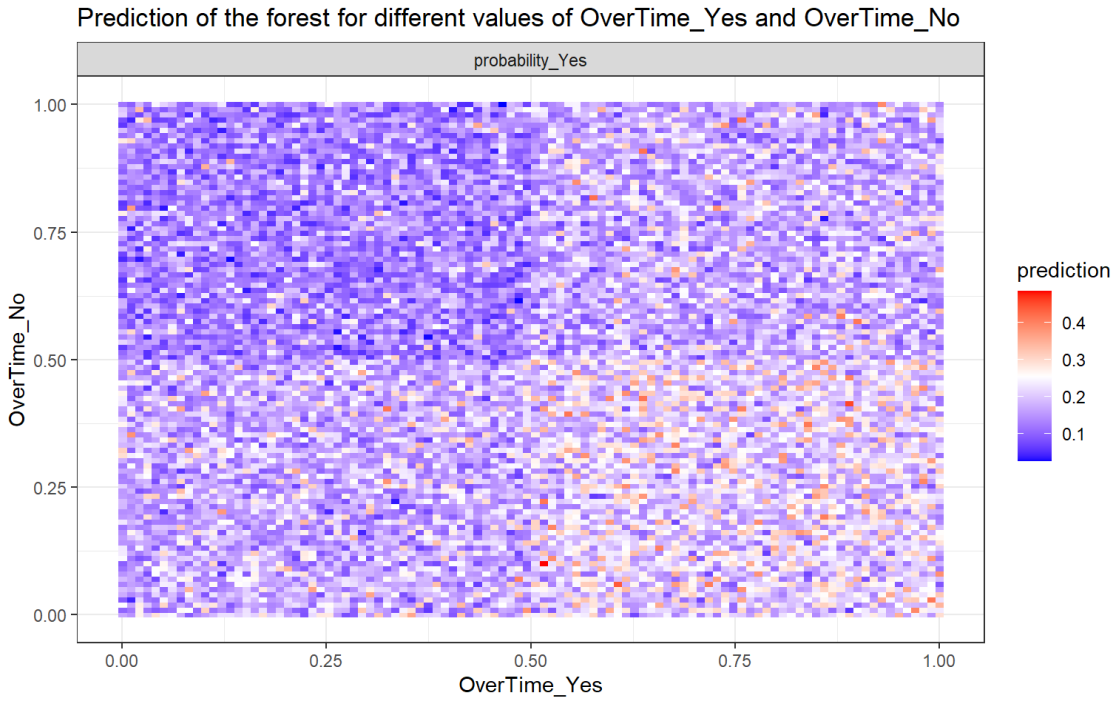

Exhibit 19

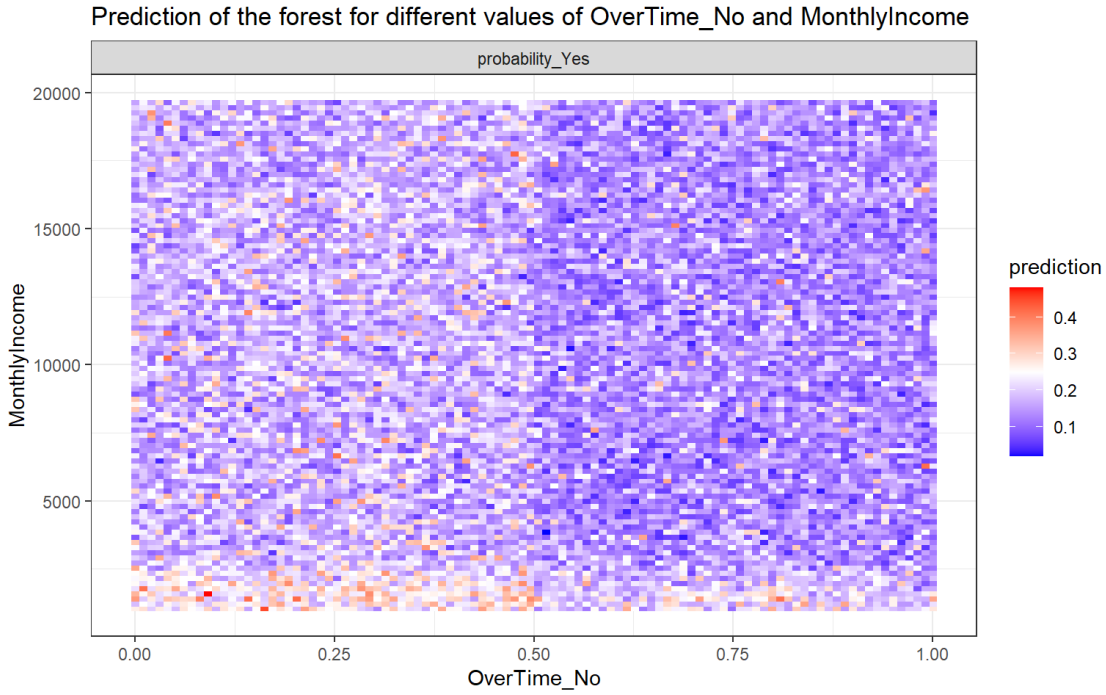

Exhibit 20

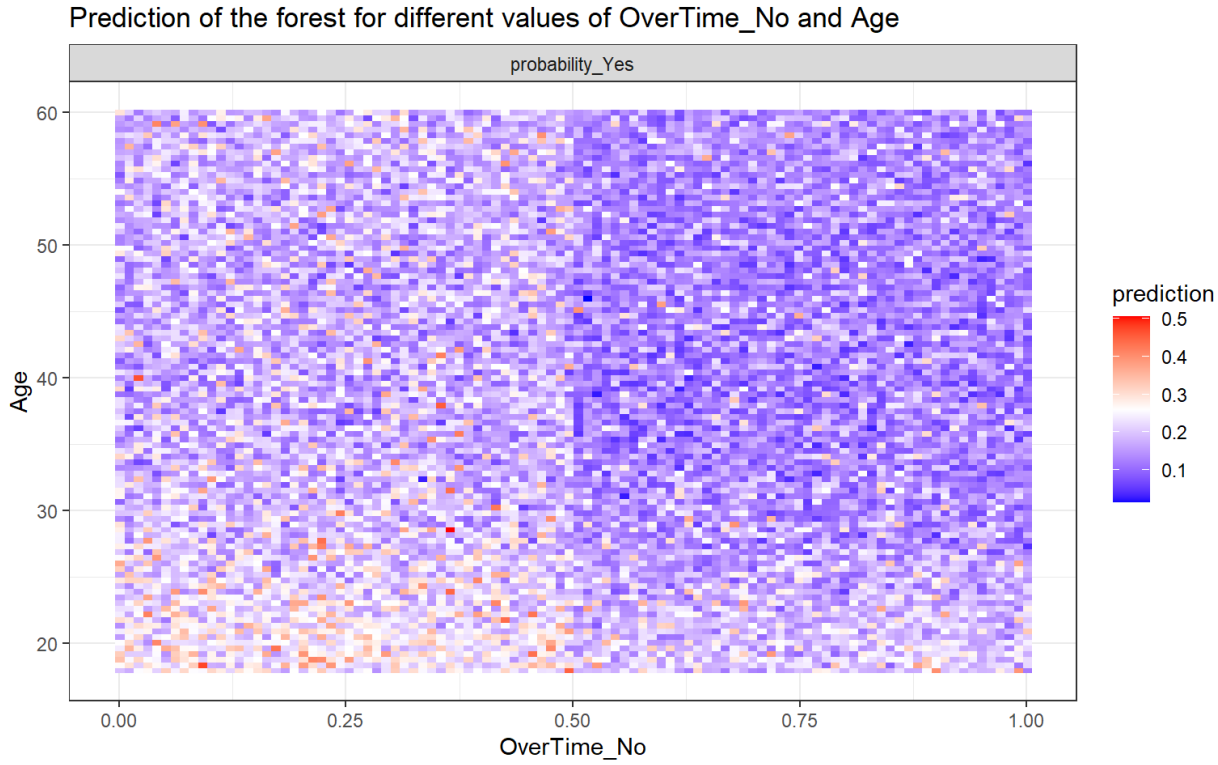

Exhibit 21

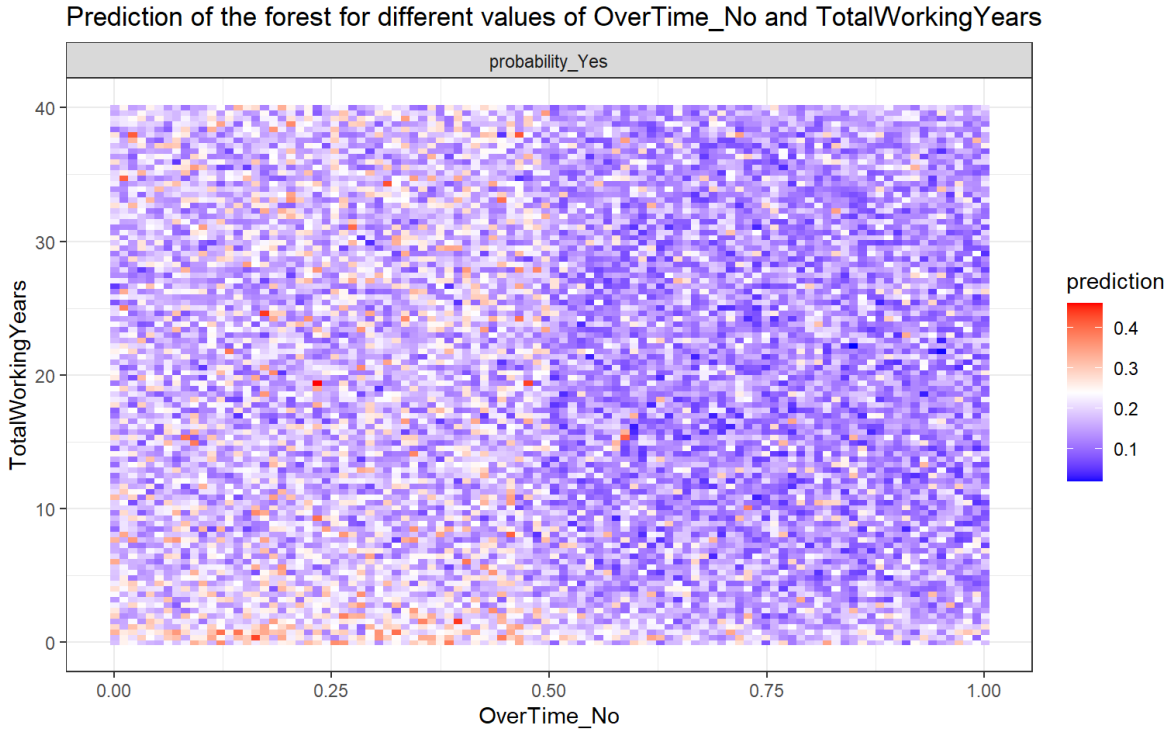

Exhibit 22

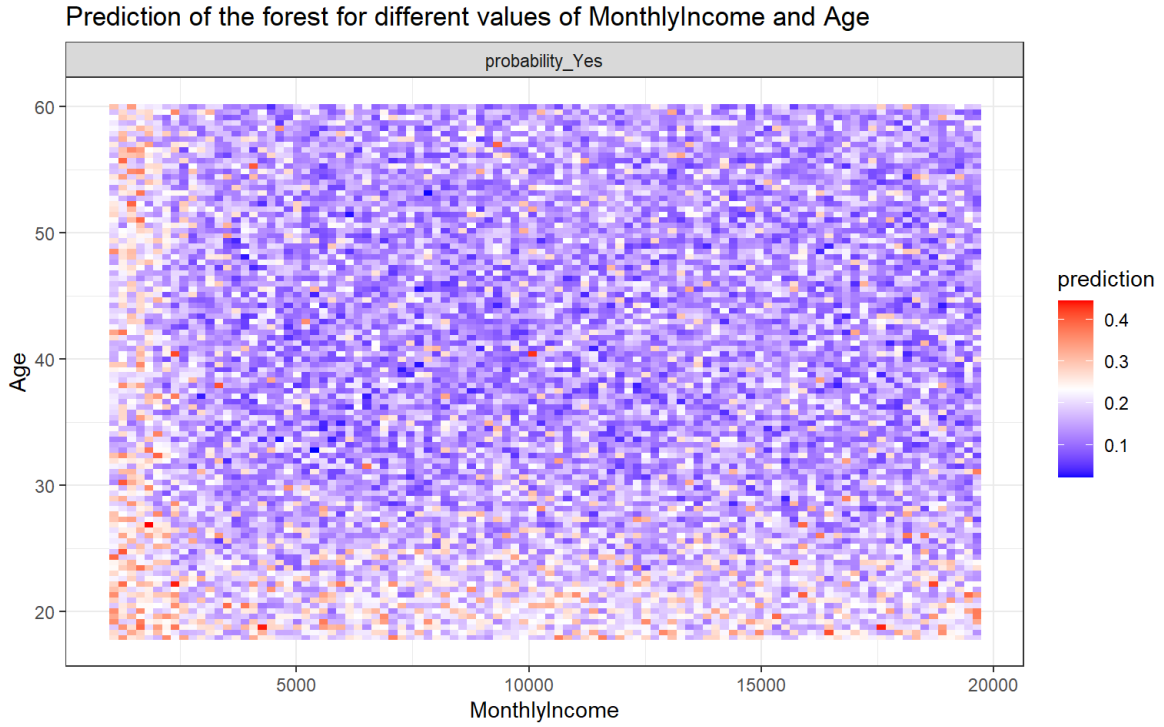

Exhibit 23

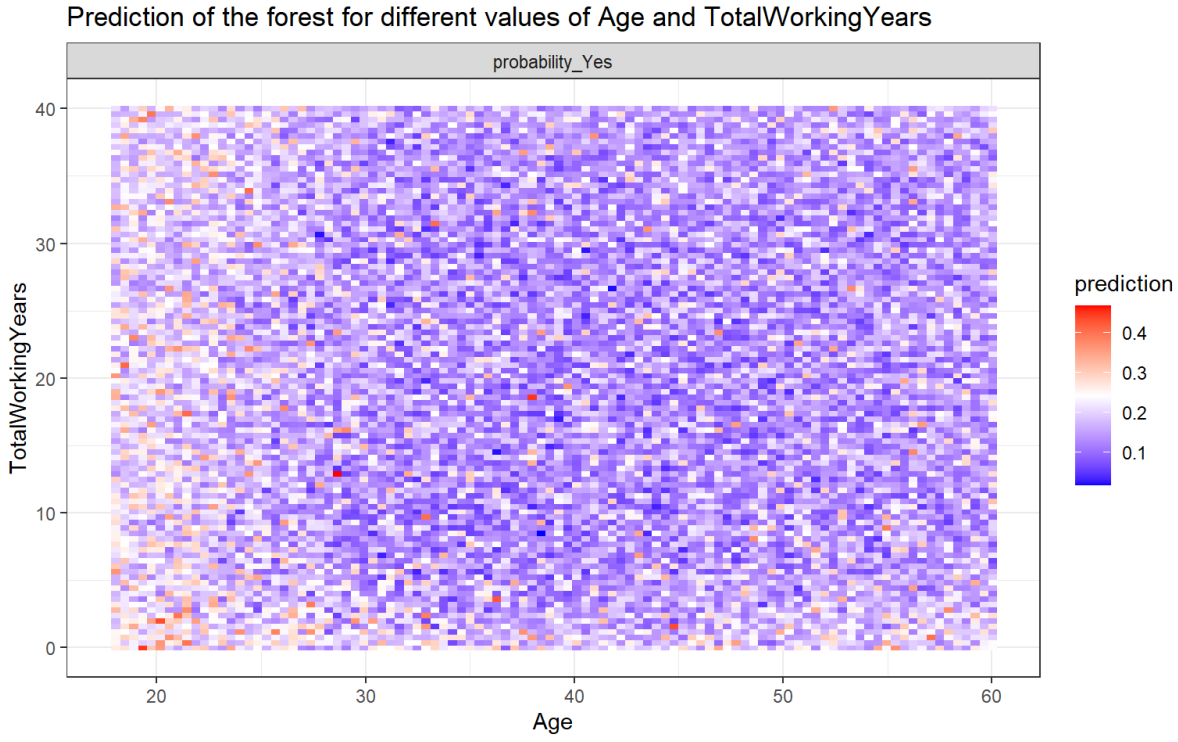

Exhibit 24: Project Code

````
## Intro
### Set-up
Load packages:
```{r message=FALSE, warning=FALSE}
if (!require(readxl)) install.packages("readxl")
library(readxl)
if (!require(tidyverse)) install.packages("tidyverse")
library(tidyverse)
if (!require(ggplot2)) install.packages("ggplot2")
library(ggplot2)
# theme_set(theme_classic())
theme_set(theme_bw())
if (!require(ggExtra)) install.packages("ggExtra")
library(ggExtra)


if (!require(e1071)) install.packages("e1071")
library(e1071)
if (!require(fastDummies)) install.packages("fastDummies")
library(fastDummies)
if (!require(caret)) install.packages("caret")
library(caret)
if (!require(tree)) install.packages('tree')
library(tree)
if (!require(rpart)) install.packages('rpart')
library(rpart)
library(rpart.plot)
if (!require(progress)) install.packages("progress")
library(progress)
if (!require(randomForest)) install.packages("randomForest")
library(randomForest)
if (!require(randomForestExplainer))
install.packages("randomForestExplainer")
library(randomForestExplainer)
if (!require(pROC)) install.packages('pROC')
library(pROC)
```

### Import Data
```{R}
raw_data <-
  read.csv(
    "../data/WA_Fn-UseC_-HR-Employee-Attrition.csv",
    stringsAsFactors = T,
    na.strings = '?'
````
````

```
  )

names(raw_data)[names(raw_data) == 'ï..Age'] <- 'Age'

glimpse(raw_data)
```

Preprocess the data
```{R}

tmp_Attrition <- raw_data$Attrition

data_droped_col <-
  subset(raw_data,
         select = -c(Attrition, EmployeeCount, StandardHours,
Over18))

data_dummied <- dummy_cols(data_droped_col)

data_processed <-
  data_dummied[, sapply(data_dummied, class) != "factor"]

colnames(data_processed) <- gsub(" ", "", colnames(data_processed))
colnames(data_processed) <- gsub("-", "", colnames(data_processed))
colnames(data_processed) <- gsub("&", "", colnames(data_processed))
colnames(data_processed) <- gsub("`", "", colnames(data_processed))

data_processed$Attrition <- tmp_Attrition

glimpse(data_processed)
```

```{R}
set.seed(10)

split <- sample(nrow(data_processed), nrow(data_processed) * 0.75)

data_train <- data_processed[split,]
data_test <- data_processed[-split,]

```

Understanding the data
Histograms
```{r}
raw_data %>%
  ggplot(aes(x = Attrition)) +
```

```
  geom_bar()

raw_data %>%
  ggplot(aes(x = Age)) +
  geom_histogram(binwidth = 10, aes(fill = Attrition))

raw_data %>%
  ggplot(aes(x = MonthlyIncome)) +
  geom_histogram(binwidth = 1000, aes(fill = Attrition))

raw_data %>%
  ggplot(aes(x = OverTime)) +
  geom_bar(aes(fill = Attrition))

raw_data %>%
  ggplot(aes(Department)) +
  geom_bar(aes(fill = Attrition))

raw_data %>%
  ggplot(aes(Gender)) +
  geom_bar(aes(fill = Attrition))
```

Scatterplot

```{r message=FALSE, warning=FALSE}

incomeAge <- raw_data %>%
  ggplot(aes(MonthlyIncome, Age)) +
  geom_point(aes(color = Attrition)) +
  geom_smooth(method = "gam", se = F) +
  theme(legend.position = "bottom")
incomeAge %>%  ggMarginal(type = "histogram", groupFill = T)

```

```{r message=FALSE, warning=FALSE}

workyear <- raw_data %>%
  ggplot(aes(TotalWorkingYears, YearsAtCompany)) +
  geom_point(aes(color = Attrition)) +
  geom_smooth(method = "gam", se = F) +
  theme(legend.position = "bottom")
workyear %>%  ggMarginal(type = "histogram", groupFill = T)

```

Modeling

Decision Tree
build a model with a training set
```{r}
training_model <- rpart(Attrition ~ .,
                        data = data_train,
                        method = "class")

rpart.plot(training_model)
```

```{r}
dt_table <-
  table(
    ifelse(predict(training_model, data_test)[, 2] > 0.5 , "Yes",
"No"),
    data_test$Attrition,
    dnn = c("predicted", "actual")
  )
confusionMatrix(dt_table)
```

k-fold Cross-validation, cp with 0
```{r}
full_tree <- rpart(
  Attrition ~ .,
  data = data_train,
  method = "class",
  control = rpart.control(cp = 0) #cp=0, more complex, let the tree
grow
)
rpart.plot(full_tree)
```

```{r}
printcp(full_tree)   # xerror, xstd - cross validation results
```

```{r}
plotcp(full_tree)
```

```{r}
min_xerror <-
  full_tree$cptable[which.min(full_tree$cptable[, "xerror"]), ]
min_xerror

# prune tree with minimum cp value
```

```
min_xerror_tree <- prune(full_tree, cp = min_xerror[1])
rpart.plot(min_xerror_tree)
```

```{r}
bp_tree <- min_xerror_tree
data_test$ct_pred_prob <- predict(bp_tree, data_test)[, 2]
data_test$ct_pred_class <- ifelse(data_test$ct_pred_prob > 0.5,
"Yes", "No")

dt_table <-
  table(data_test$ct_pred_class,
        data_test$Attrition,
        dnn = c("predicted", "actual"))
confusionMatrix(dt_table)
```

Random Forest
baseline forest
```{r}
treeCount <- 1000

data.forest <- randomForest(
  Attrition ~ .,
  data = data_train,
  ntree = treeCount,
  importance = T,
  localImp = T
)
plot(data.forest)
```

testing hyper peramiter: mtry
```{r}
data.forest <- randomForest(
  Attrition ~ .,
  data = data_train,
  ntree = treeCount,
  mtry = ncol(data_processed) - 1,
  importance = T,
  localImp = T
)
data.forest
#+ fig.width=11, fig.height=8
plot(data.forest)
```

```{r fig.height=10, fig.width=10}
data.frame(importance(data.forest))
varImpPlot(data.forest)
```

```{r}
data.forest.pred <- predict(data.forest, data_test, typr = "class")
rf_confMtx <-
  table(data.forest.pred,
        data_test$Attrition,
        dnn = c("predicted", "actual"))
confusionMatrix(rf_confMtx)
```

Brute force search for best mtry
```{r}
pb <- progress_bar$new(format = ":elapsedfull [:bar] :current/:total
(:percent)",
                       total = (ncol(data_train) - 1))

mtry.list <- c()
data.forest.final <- NULL
max <- 0
for (i in 1:(ncol(data_train) - 1)) {
  data.forest.dummy <- randomForest(
    Attrition ~ .,
    data = data_train,
    ntree = treeCount,
    mtry = i,
    importance = T,
    localImp = T
  )
  val <- predict(data.forest.dummy, data_test, type = "class")
  score <- mean(val == data_test$Attrition)
  if (score >= max) {
    data.forest.final <- data.forest.dummy
    max <- score
  }
  mtry.list[i] <- score
  pb$tick()
}

plot(mtry.list)

data.forest.final
plot(data.forest.final)
```

```{r fig.height=10, fig.width=10}
data.frame(importance(data.forest.final))

varImpPlot(data.forest.final)

data_test$rf_pred_prob <-
  predict(data.forest.final, data_test, type = "prob")[, 2]
data_test$rf_pred_class <- predict(data.forest.final, data_test)

rf_confMtx <-
  table(data_test$rf_pred_class,
        data_test$Attrition,
        dnn = c("predicted", "actual"))
confusionMatrix(rf_confMtx)
```

Plot forest predict
Plotting the best forest's predictions in associate with top 5 most important independent variables
```{r message=FALSE, warning=FALSE}
imp <- data.frame(importance(data.forest.final))

imp <- imp[order(-imp$MeanDecreaseAccuracy), ]

impName <- row.names(imp)

impNameComp <- combn(impName[1:5], 2)

plots <- apply(impNameComp, MARGIN = 2, FUN = function(i) {
  plot_predict_interaction(data.forest.final, data_test, i[1], i[2])
})

invisible(capture.output(print(plots)))
```

```{r eval=FALSE, include=FALSE}
# auto generate random forest model report
explain_forest(
  data.forest.final,
  path = paste0(getwd(), "/HR_Analysis_forest_explained.html"),
  interactions = TRUE,
  no_of_pred_plots = 10,
  data = data_test
)
```

```
### SVM
#### Base line SVM model
```{R}
svm_1 <-
  svm(
    Attrition ~ .,
    data = data_train,
    method = "C-classification",
    kernal = "radial"
  )

summary(svm_1)
```

```{R}
svm_1_pred <- predict(svm_1, data_test)

xtab_svm_1 <-
  table(svm_1_pred, data_test$Attrition, dnn = c("predicted",
"actual"))

confusionMatrix(xtab_svm_1)
```
Tuning SVM model
```{R}

svm_tune <-
  tune.svm(
    Attrition ~ .,
    data = data_train,
    kernel = "radial",
    cost = 10 ^ (-5:4),
    gamma = c(seq(1, 10, 4) %o% 10 ^ (-4:1))
  )

svm_tune$performances
```

```{R}
best_svm_mod <- svm_tune$best.model

data_test$svm_pred_class <- predict(best_svm_mod, data_test)
data_test$svm_dv <-
  as.numeric(attr(
    predict(best_svm_mod, data_test, decision.value = TRUE),
    "decision.values"
```
```

```
  ))

xtab_svm_best <-
  table(data_test$svm_pred_class,
        data_test$Attrition,
        dnn = c("predicted", "actual"))

confusionMatrix(xtab_svm_best)
```

Evaluation
Decision Tree
```{r}
confusionMatrix(dt_table)
```
Random Forest
```{r}
confusionMatrix(rf_confMtx)
```
SVM
```{r}
confusionMatrix(xtab_svm_best)
```

ROC & AUC
```{r message=FALSE, warning=FALSE}
ct_roc <- roc(data_test$Attrition, data_test$ct_pred_prob, auc =
TRUE)
rf_roc <- roc(data_test$Attrition, data_test$rf_pred_prob, auc =
TRUE)
svm_roc <- roc(data_test$Attrition, data_test$svm_dv, auc = TRUE)

plot(ct_roc, print.auc = TRUE, col = "blue")
plot(rf_roc, print.auc = TRUE, print.auc.y = .4, col = "green", add =
TRUE)
plot(svm_roc, print.auc = TRUE, print.auc.y = .3, col = "black", add
= TRUE)
```