Data Mining Project: HR Employees

David Medina Hernandez

03/2018

```
library(ggplot2)
library(caret) # setting seeds
```

```
## Loading required package: lattice
```

```
library(MASS) # LDA
library(tree)
library(randomForest)
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(cowplot) # multiple plots in one window
```

```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:ggplot2':
##
##     ggsave
```

```
library("ggthemes")
```

```
##
## Attaching package: 'ggthemes'
```

```
## The following object is masked from 'package:cowplot':
##
##     theme_map
```

```
library(knitr)
```

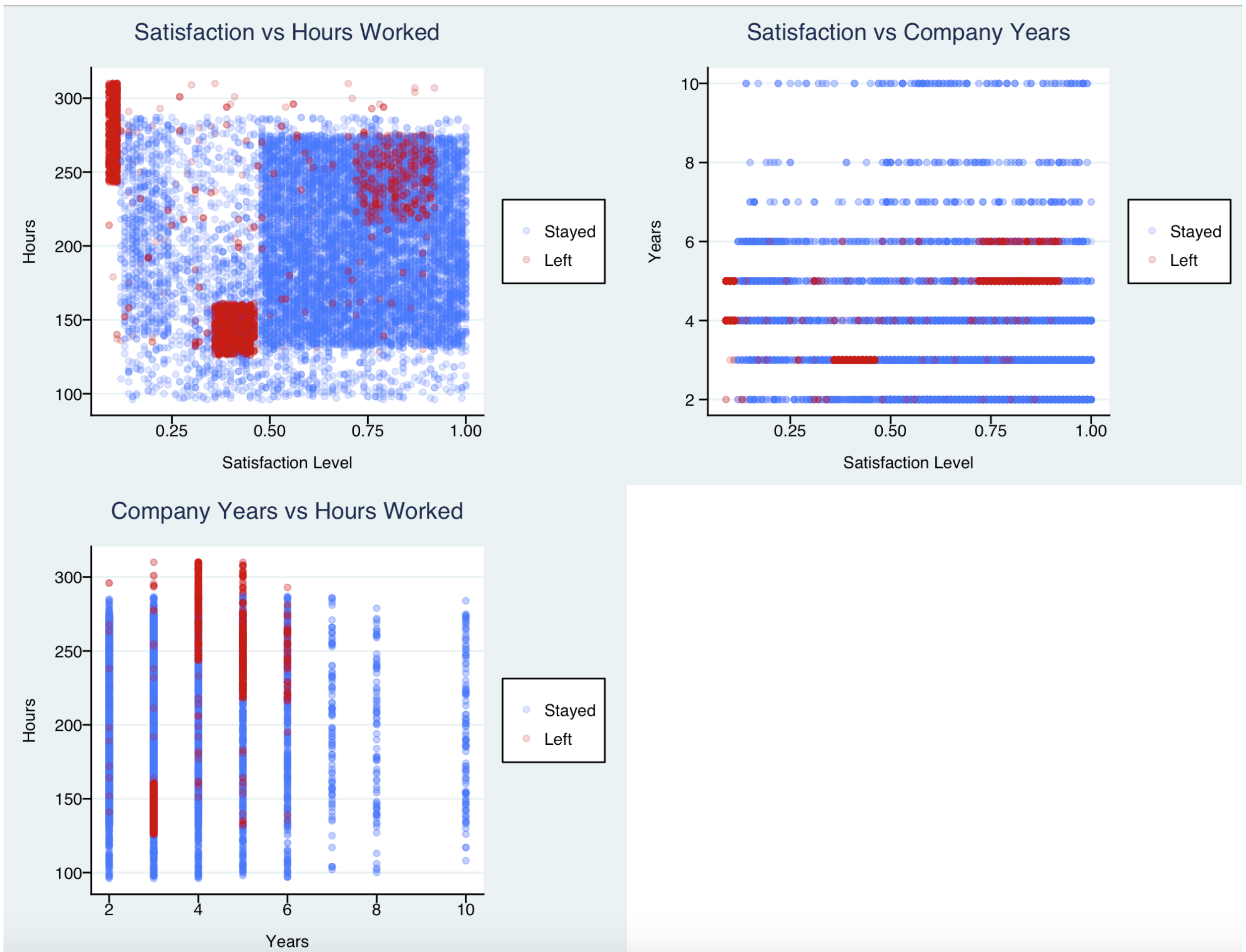
```
# This data mining project compares the accuracy of different models
# Predicting if an employee will leave the company using a Kaggle data set
hr_ds <- read.csv("/Users/davidmedina/Desktop/Current job forms/coding samples/hr/kaggle_HR/hr.csv")
summary(hr_ds)
```

```
##  satisfaction_level last_evaluation  number_project  average_montly_hours
##  Min.   :0.0900     Min.   :0.3600   Min.   :2.000   Min.   : 96.0
##  1st Qu.:0.4400     1st Qu.:0.5600   1st Qu.:3.000   1st Qu.:156.0
##  Median :0.6400     Median :0.7200   Median :4.000   Median :200.0
##  Mean   :0.6128     Mean   :0.7161   Mean   :3.803   Mean   :201.1
##  3rd Qu.:0.8200     3rd Qu.:0.8700   3rd Qu.:5.000   3rd Qu.:245.0
##  Max.   :1.0000     Max.   :1.0000   Max.   :7.000   Max.   :310.0
##
```

```
##   time_spend_company Work_accident         left
##   Min.   : 2.000     Min.   :0.0000   Min.   :0.0000
##   1st Qu.: 3.000     1st Qu.:0.0000   1st Qu.:0.0000
##   Median : 3.000     Median :0.0000   Median :0.0000
##   Mean   : 3.498     Mean   :0.1446   Mean   :0.2381
##   3rd Qu.: 4.000     3rd Qu.:0.0000   3rd Qu.:0.0000
##   Max.   :10.000     Max.   :1.0000   Max.   :1.0000
##
##   promotion_last_5years       department      salary
##   Min.   :0.00000       sales      :4140   high  :1237
##   1st Qu.:0.00000       technical  :2720   low   :7316
##   Median :0.00000       support    :2229   medium:6446
##   Mean   :0.02127       IT         :1227
##   3rd Qu.:0.00000       product_mng: 902
##   Max.   :1.00000       marketing  : 858
##                         (Other)    :2923
```

```
colnames(hr_ds) <- tolower(colnames(hr_ds))
hr_ds$left <- factor(hr_ds$left, levels = 0:1, labels = c("Stayed", "Left"))
hr_ds$work_accident <- factor(hr_ds$work_accident,
                              levels = 0:1, labels = c("no", "yes"))
hr_ds$promotion_last_5years <- factor(hr_ds$promotion_last_5years,
                                      levels = 0:1, labels = c("no", "yes"))

# cuts determined from plots generated later
hr_ds$sat_cut <- cut(hr_ds$satisfaction_level, c(0, .13, .34, .50, .70, .95, Inf))
hr_ds$le_cut <- cut(hr_ds$last_evaluation, c(0, .60, .75, Inf))
hr_ds$amh_cut <- cut(hr_ds$average_montly_hours, c(0, 170, 210, Inf))

# second (unscaled) data frame created for plotting/visualization purposes
hr_ds2 <- hr_ds
# scaled variables
hr_ds[, c(1:5)] <- scale(hr_ds[, c(1:5)])

# visualization
p1 <- ggplot(data = hr_ds2, aes(x = satisfaction_level ,y = average_montly_hours,
                                color = left)) + geom_point(alpha = .2) +
  labs(x = "\nSatisfaction Level", y = "Hours\n",
       title = "Satisfaction vs Hours Worked\n") +
  scale_color_manual(name = NULL, values = c("royalblue1", "red3")) +
  theme_stata() +
  theme(axis.text.y = element_text(angle = 0)) +
  theme(legend.position = "right")

p2 <- ggplot(data = hr_ds2, aes(x = satisfaction_level,y = time_spend_company,
                                color = left)) + geom_point(alpha = .2) +
  labs(x = "\nSatisfaction Level", y = "Years\n",
       title = "Satisfaction vs Company Years\n") +
  scale_color_manual(name = NULL, values = c("royalblue1", "red3")) +
  theme_stata() +
  theme(axis.text.y = element_text(angle = 0)) +
  theme(legend.position = "right")
```

```r
p3 <- ggplot(data = hr_ds2, aes(x = time_spend_company,
                                y = average_montly_hours, color = left)) +
  geom_point(alpha = .2) +
  labs(x = "\nYears", y = "Hours\n",
       title = "Company Years vs Hours Worked\n") +
  scale_color_manual(name = NULL, values = c("royalblue1", "red3")) +
  theme_stata() +
  theme(axis.text.y = element_text(angle = 0)) +
  theme(legend.position = "right")

cuts_plot <- plot_grid(p1, p2, p3, ncol = 2)
```




```r
# regression model results
set.seed(12345)
in_train <- createDataPartition(y = hr_ds$left,
                                p = 3 / 4, list = FALSE)
training <- hr_ds[in_train, ]
testing <- hr_ds[-in_train, ]

# not scaled
training_ns <- hr_ds2[in_train, ]
testing_ns <- hr_ds2[-in_train, ]
```

```r
# linear regression
lm1 <- lm(left ~ . - satisfaction_level - last_evaluation -
            average_montly_hours, data = training)
```

```
## Warning in model.response(mf, "numeric"): using type = "numeric" with a
## factor response will be ignored

## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors
```

```r
y_hat_ols <- predict(lm1, newdata = testing)
z_ols <- as.integer(y_hat_ols > 0.5)
(ols_table <- table(testing$left, z_ols))
```

```
##         z_ols
##             1
##   Stayed 2857
##   Left    892
```

```r
(accuracy_ols <- ols_table[2] / sum(ols_table))
```

```
## [1] 0.2379301
```

```r
# logit with cuts
logit <- glm(left ~ . - sat_cut - le_cut - amh_cut, data = training,
             family = binomial(link = "logit"))
y_hat_logit <- predict(logit, newdata = testing, type = "response")
z_logit <- as.integer(y_hat_logit > 0.5)
(logit_table <- table(testing$left, z_logit))
```

```
##         z_logit
##             0    1
##   Stayed 2641  216
##   Left    589  303
```

```r
(accuracy_logit <- sum(diag(logit_table)) / sum(logit_table))
```

```
## [1] 0.7852761
```

```r
# logit without cuts
logit2 <- glm(left ~ . - satisfaction_level - last_evaluation -
                average_montly_hours, data = training,
              family = binomial(link = "logit"))
y_hat_logit2 <- predict(logit2, newdata = testing, type = "response")
z_logit2 <- as.integer(y_hat_logit2 > 0.5)
(logit_table2 <- table(testing$left, z_logit2))
```

```
##         z_logit2
##             0    1
##   Stayed 2698  159
##   Left    275  617
```

```r
(accuracy_logit2 <- sum(diag(logit_table2)) / sum(logit_table2))
```

```
## [1] 0.8842358
```

```r
# linear discriminant analysis
LDA <- lda(left ~ . - satisfaction_level - last_evaluation -
             average_montly_hours, data = training)
y_hat_LDA <- predict(LDA, newdata = testing)
```

```r
z_LDA <- y_hat_LDA$class
(LDA_table <- table(testing$left, z_LDA))
```

```
##         z_LDA
##          Stayed Left
##   Stayed   2674  183
##   Left      303  589
```

```r
(accuracy_LDA <- sum(diag(LDA_table)) / sum(LDA_table))
```

```
## [1] 0.8703654
```

```r
# Tree Based Model Results

# basic tree model
out <- tree(left ~ . - satisfaction_level - last_evaluation -
              average_montly_hours, data = training)
new_out <- cv.tree(out, FUN = prune.misclass)
# pruning tree
best_model <- prune.tree(out, best = 8)
pred_ptree <- predict(best_model, newdata = testing, type = "class")
tree_table <- table(testing$left, pred_ptree)
(accuracy_tree <- sum(diag(tree_table)) / sum(tree_table))
```

```
## [1] 0.9527874
```

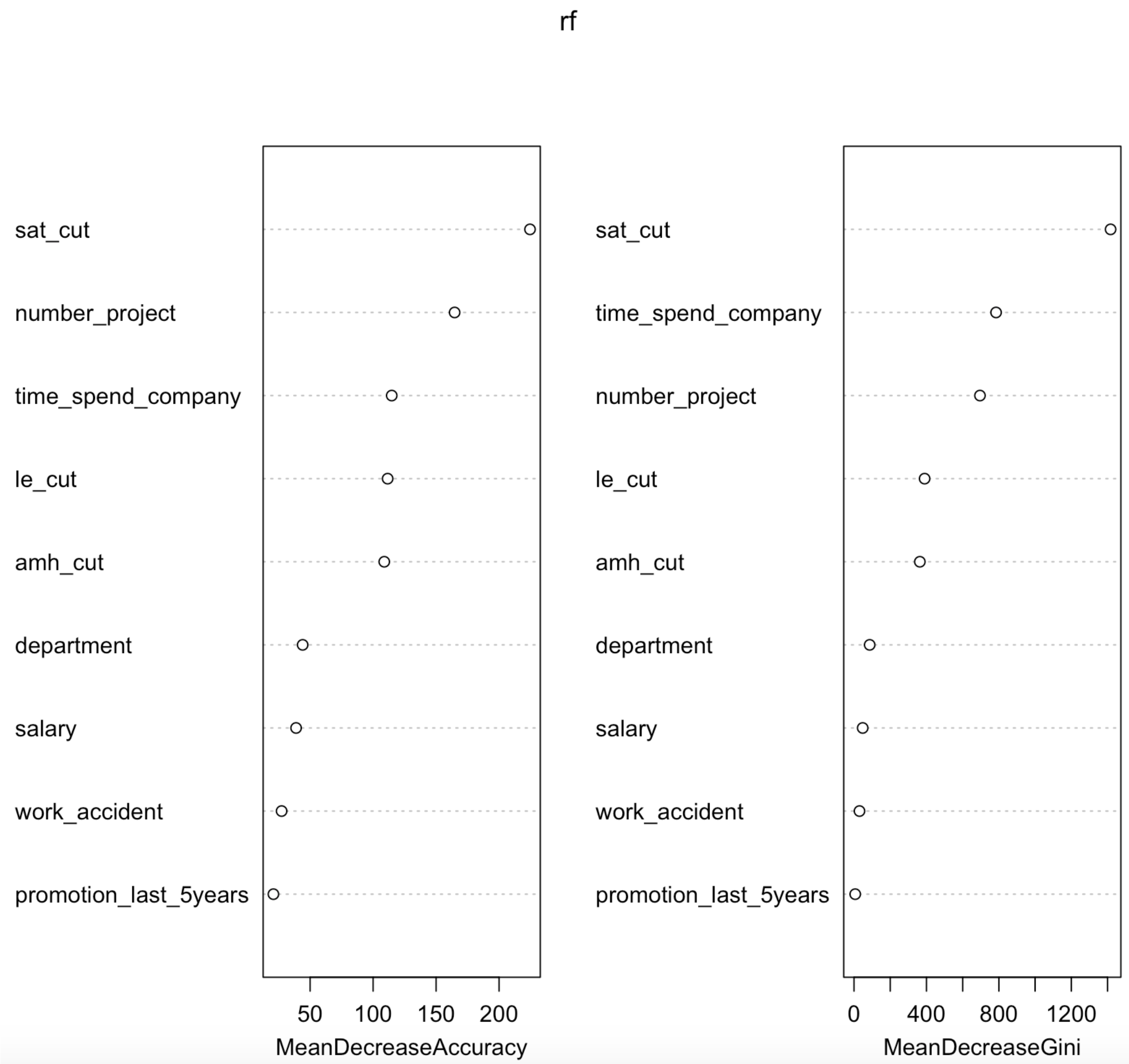
```r
# random forest
rf <- randomForest(left ~ . - satisfaction_level - last_evaluation -
                     average_montly_hours, data = training, importance = TRUE)
pred_rf <- predict(rf, newdata = testing, type = "class")
(rf_table <- table(testing$left, pred_rf))
```

```
##         pred_rf
##          Stayed Left
##   Stayed   2844   13
##   Left       67  825
```

```r
(accuracy_rf <- sum(diag(rf_table)) / sum(rf_table))
```

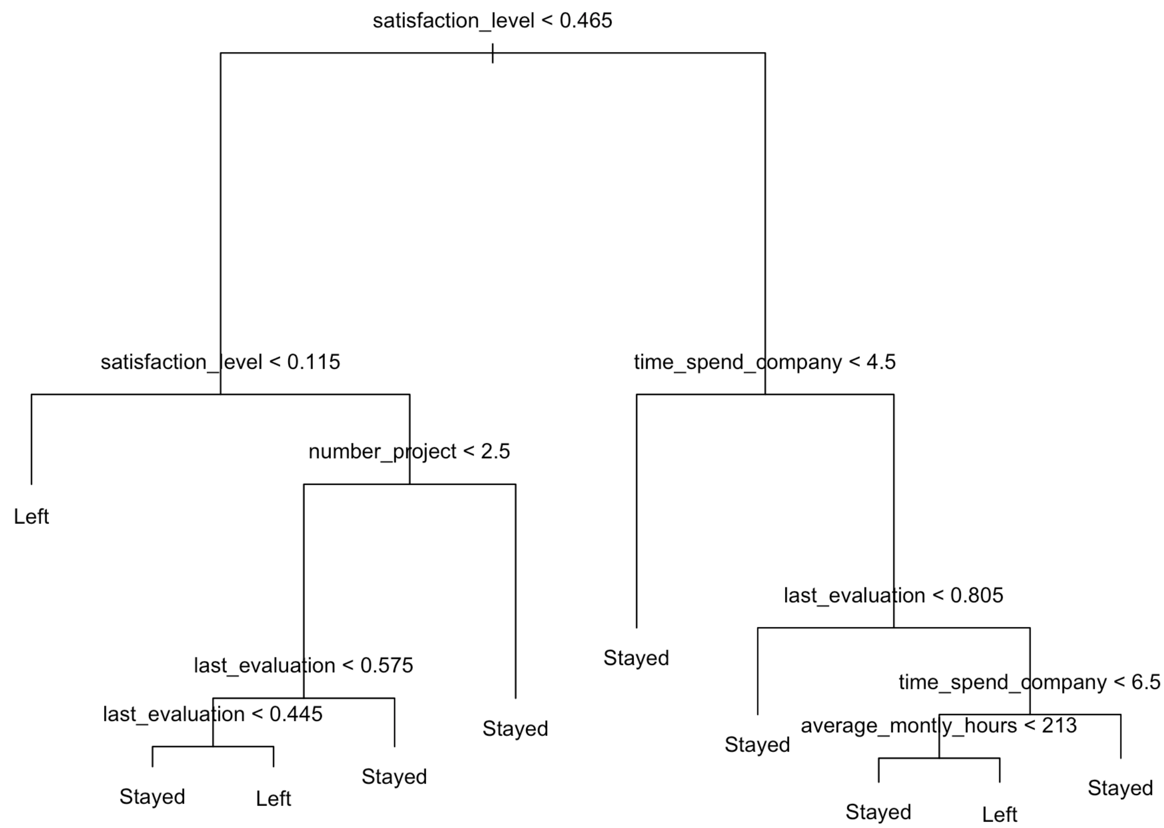
```
## [1] 0.978661
```

```r
# varImpPlot(rf)
```




```
# visual tree
out_ns <- tree(left ~ . - sat_cut - le_cut - amh_cut, data = training_ns)
# plot(out_ns); text(out_ns, pretty = 0)
pred_tree_ns <- predict(out_ns, newdata = testing_ns, type = "class")
tree_table_ns <- table(testing_ns$left, pred_tree_ns)
(accuracy_one_tree <- sum(diag(tree_table_ns)) / sum(tree_table_ns))
```

```
## [1] 0.9666578
```




```
# below is a summary of the accuracy for all algorithms used:

names_model <- c("linear prob", "logit no cut", "logit with cut", "LDA",
                 "prune tree", "random forest", "single tree")

accuracy_num <- c(accuracy_ols, accuracy_logit, accuracy_logit2,
                  accuracy_LDA, accuracy_tree, accuracy_rf,
                  accuracy_one_tree )

accuracy_table <- cbind(names_model, accuracy = round(accuracy_num, digits = 4))
```

Model	Accuracy
linear prob	0.2379
logit no cut	0.7853
logit with cut	0.8842
LDA	0.8704
prune tree	0.9528
random forest	0.9787
single tree	0.9667