

People Analytics Using R – Employee Churn

Step 1. Define The Goal

My hypothetical company wants to apply data analytics principles and steps to deal with a critical HR issue: employee churn. It realizes that when good people leave, it costs far more to replace them than providing them with some incentives to keep them. So, it would like to be data-driven in the HR decisions it makes with respect to employee retention.

The following questions are among the ones they would like answered:

1. What proportion of our people is leaving?
2. Where is it occurring?
3. How does Age and Length of Service affect termination?
4. What, if anything, else contributes to it?
5. Can we predict future terminations?
6. If so, how well can we predict?

Step 2 – Collect and Manage the Data

Often the data to analyze the problem starts with what is currently readily available. After some initial prototyping of predictive models, ideas surface for additional data collection to further refine the model. Since this is first stab at this, the organization uses only what is readily available.

After consulting with their HRIS staff, they found that they have access to the following information:

- EmployeeID
- Record Date
- Birth Date
- Original Hire Date
- Termination Date (if terminated)

- Age
- Length of Service
- City
- Department
- Job title
- Store Name
- Gender
- termination reason
- termination type (voluntary or involuntary)
- Status Year – year of data
- Status – ACTIVE or TERMINATED during Status year
- Business Unit – either Stores or Head Office

The company found out that they have 10 years of good data – from 2006 to 2015. It wants to use 2006-2014 as training data and use 2015 as the data to test on. The data consists of

- a snapshot of all active employees at the end of each of those years combined with
- terminations that occurred during each of those years.

Therefore, each year will have records that have either a status of ‘active’ or ‘terminated’. Of the above information items listed, the ‘STATUS’ one is the ‘dependent’ variable. This is the category to be predicted. The others are independent variables. They are the potential predictors.

First Look at The Data – The Structure

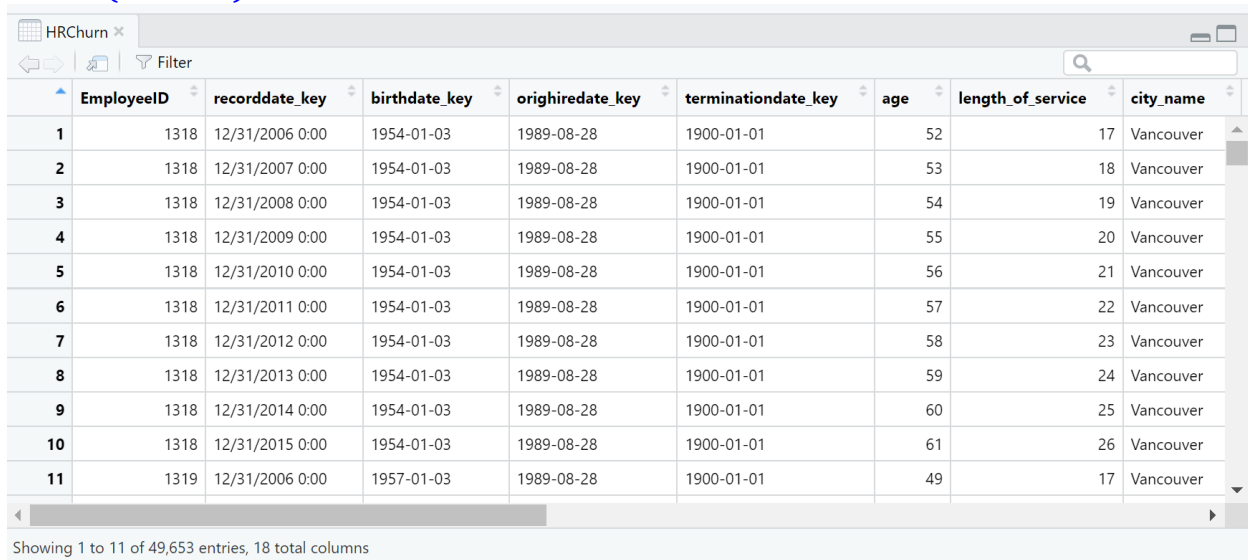
```
# Load an R data frame
> HRChurn <- read.csv("C://Users/alire/Downloads/Datasets-Tutorial-People-Analytics/HRChurn.csv")
> MYdataset <- HRChurn

# Display the internal structure
> str(MYdataset)
```

```
> str(MYdataset)
'data.frame': 49653 obs. of 18 variables:
 $ EmployeeID      : int  1318 1318 1318 1318 1318 1318 1318 1318 ...
 $ recorddate_key   : chr   "12/31/2006 0:00" "12/31/2007 0:00" "12/31/2008 0:00" "12/31/2009 0:00" ...
 $ birthdate_key    : chr   "1954-01-03" "1954-01-03" "1954-01-03" "1954-01-03" ...
 $ orighiredate_key : chr   "1989-08-28" "1989-08-28" "1989-08-28" "1989-08-28" ...
 $ terminationdate_key: chr   "1900-01-01" "1900-01-01" "1900-01-01" "1900-01-01" ...
 $ age             : int    52 53 54 55 56 57 58 59 60 61 ...
 $ length_of_service : int    17 18 19 20 21 22 23 24 25 26 ...
 $ city_name        : chr   "Vancouver" "Vancouver" "Vancouver" "Vancouver" ...
 $ department_name  : chr   "Executive" "Executive" "Executive" "Executive" ...
 $ job_title         : chr   "CEO" "CEO" "CEO" "CEO" ...
 $ store_name        : int    35 35 35 35 35 35 35 35 35 35 ...
 $ gender_short      : chr   "M" "M" "M" "M" ...
 $ gender_full       : chr   "Male" "Male" "Male" "Male" ...
 $ termreason_desc   : chr   "Not Applicable" "Not Applicable" "Not Applicable" "Not Applicable" ...
 $ termtype_desc     : chr   "Not Applicable" "Not Applicable" "Not Applicable" "Not Applicable" ...
 $ STATUS_YEAR       : int    2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 ...
 $ STATUS            : chr   "ACTIVE" "ACTIVE" "ACTIVE" "ACTIVE" ...
 $ BUSINESS_UNIT     : chr   "HEADOFFICE" "HEADOFFICE" "HEADOFFICE" "HEADOFFICE" ...
```

opening a spreadsheet-style of dataset

```
> View(HRChurn)
```



	EmployeeID	recorddate_key	birthdate_key	orighiredate_key	terminationdate_key	age	length_of_service	city_name
1	1318	12/31/2006 0:00	1954-01-03	1989-08-28	1900-01-01	52	17	Vancouver
2	1318	12/31/2007 0:00	1954-01-03	1989-08-28	1900-01-01	53	18	Vancouver
3	1318	12/31/2008 0:00	1954-01-03	1989-08-28	1900-01-01	54	19	Vancouver
4	1318	12/31/2009 0:00	1954-01-03	1989-08-28	1900-01-01	55	20	Vancouver
5	1318	12/31/2010 0:00	1954-01-03	1989-08-28	1900-01-01	56	21	Vancouver
6	1318	12/31/2011 0:00	1954-01-03	1989-08-28	1900-01-01	57	22	Vancouver
7	1318	12/31/2012 0:00	1954-01-03	1989-08-28	1900-01-01	58	23	Vancouver
8	1318	12/31/2013 0:00	1954-01-03	1989-08-28	1900-01-01	59	24	Vancouver
9	1318	12/31/2014 0:00	1954-01-03	1989-08-28	1900-01-01	60	25	Vancouver
10	1318	12/31/2015 0:00	1954-01-03	1989-08-28	1900-01-01	61	26	Vancouver
11	1319	12/31/2006 0:00	1957-01-03	1989-08-28	1900-01-01	49	17	Vancouver

Showing 1 to 11 of 49,653 entries, 18 total columns

Installing plyr abd dplyr packages

```
> install.packages("plyr")
> install.packages("dplyr")

> library(plyr)
> library(dplyr)
```

“plyr” is a package that provides a set of tools for splitting, applying, and combining large datasets. It is particularly useful for working with data frames.

“dplyr” is a grammar of data manipulation, providing a consistent set of verbs

Second Look at The Data – Data Quality

We will now assess data quality, using a few simple descriptive statistics, including the minimum value, mean, and max, and creating a number of counts.

```
# summarizing the values in each column
```

```
> summary(HRChurn)
```

```
EmployeeID      recorddate_key      birthdate_key      orighiredate_key      terminationdate_key
Min.   :1318      Length:49653      Length:49653      Length:49653      Length:49653
1st Qu.:3360      Class :character      Class :character      Class :character      Class :character
Median :5031      Mode  :character      Mode  :character      Mode  :character      Mode  :character
Mean   :4859
3rd Qu.:6335
Max.   :8336

age      length_of_service      city_name      department_name      job_title
Min.   :19.00      Min.   : 0.00      Length:49653      Length:49653      Length:49653
1st Qu.:31.00      1st Qu.: 5.00      Class :character      Class :character      Class :character
Median :42.00      Median :10.00      Mode  :character      Mode  :character      Mode  :character
Mean   :42.08      Mean   :10.43
3rd Qu.:53.00      3rd Qu.:15.00
Max.   :65.00      Max.   :26.00

store_name      gender_short      gender_full      termreason_desc      termtype_desc
Min.   : 1.0      Length:49653      Length:49653      Length:49653      Length:49653
1st Qu.:16.0      Class :character      Class :character      Class :character      Class :character
Median :28.0      Mode  :character      Mode  :character      Mode  :character      Mode  :character
Mean   :27.3
3rd Qu.:42.0
Max.   :46.0

STATUS_YEAR      STATUS      BUSINESS_UNIT
Min.   :2006      Length:49653      Length:49653
1st Qu.:2008      Class :character      Class :character
Median :2011      Mode  :character      Mode  :character
Mean   :2011
3rd Qu.:2013
Max.   :2015
```

Third Look at the Data – Generally What Is The Data Telling Us?

Earlier we had indicated that we had both active records at end of year and terminates during the year for each of 10 years going from 2006 to 2015. To have a population to model from (to differentiate ACTIVES from TERMINATES) we have to include both status types. It's useful then to get a baseline of what percent/proportion the terminates are of the entire population. It also answers our first question. Let's look at that next.

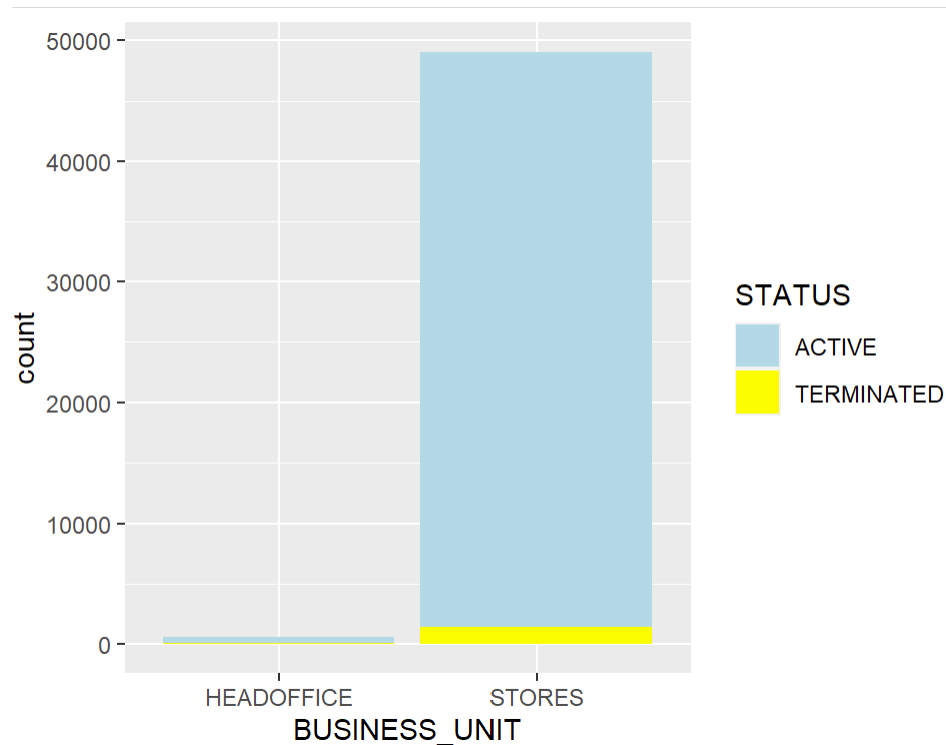
```
> library(dplyr)
> StatusCount <- as.data.frame.matrix(MYdataset %>%
  group_by(STATUS_YEAR) %>%
  select(STATUS) %>%
  table())
> StatusCount$TOTAL <- StatusCount$ACTIVE + StatusCount$TERMINATED
> StatusCount$PercentTerminated <- StatusCount$TERMINATED / (StatusCount$TOTAL) * 100
> StatusCount
```

	ACTIVE	TERMINATED	TOTAL	PercentTerminated
2006	4445	134	4579	2.926403
2007	4521	162	4683	3.459321
2008	4603	164	4767	3.440319
2009	4710	142	4852	2.926628
2010	4840	123	4963	2.478340
2011	4972	110	5082	2.164502
2012	5101	130	5231	2.485184
2013	5215	105	5320	1.973684
2014	4962	253	5215	4.851390
2015	4799	162	4961	3.265471

```
> mean(StatusCount$PercentTerminated)
[1] 2.997124
```

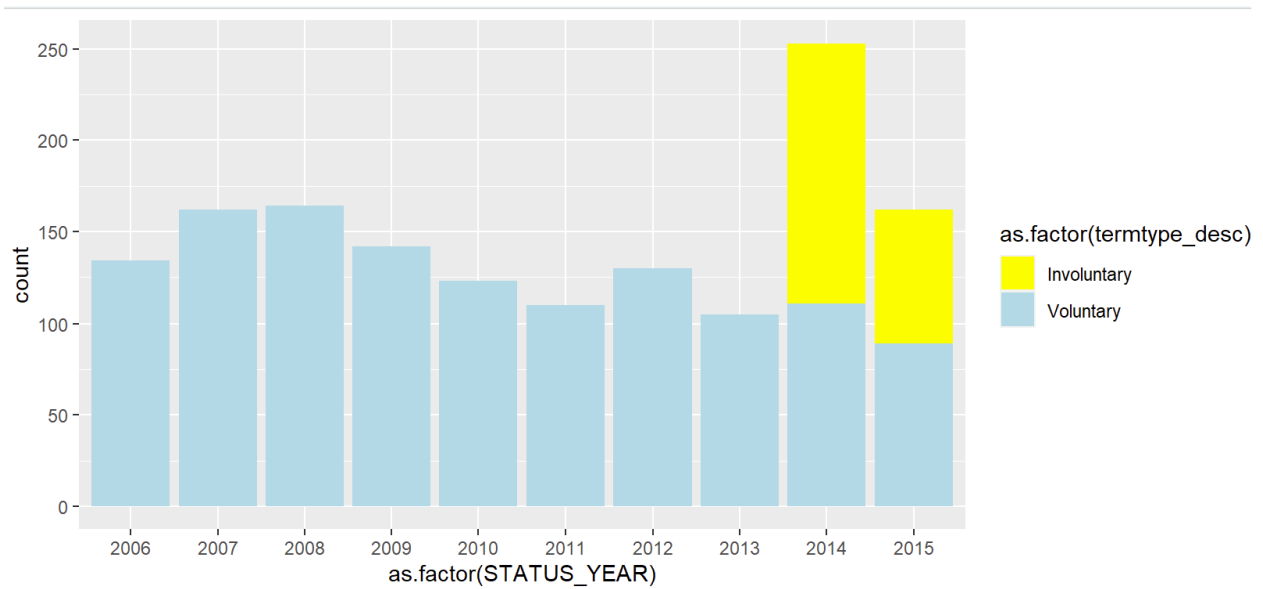
Where are the terminations occurring?

```
install.packages("ggplot2")
library(ggplot2)
ggplot(MYdataset, aes(x = BUSINESS_UNIT, fill = STATUS)) +
  geom_bar(position = position_stack()) +
  scale_fill_manual(values = c("ACTIVE" = "lightblue", "TERMINATED" = "yellow"))
))
```



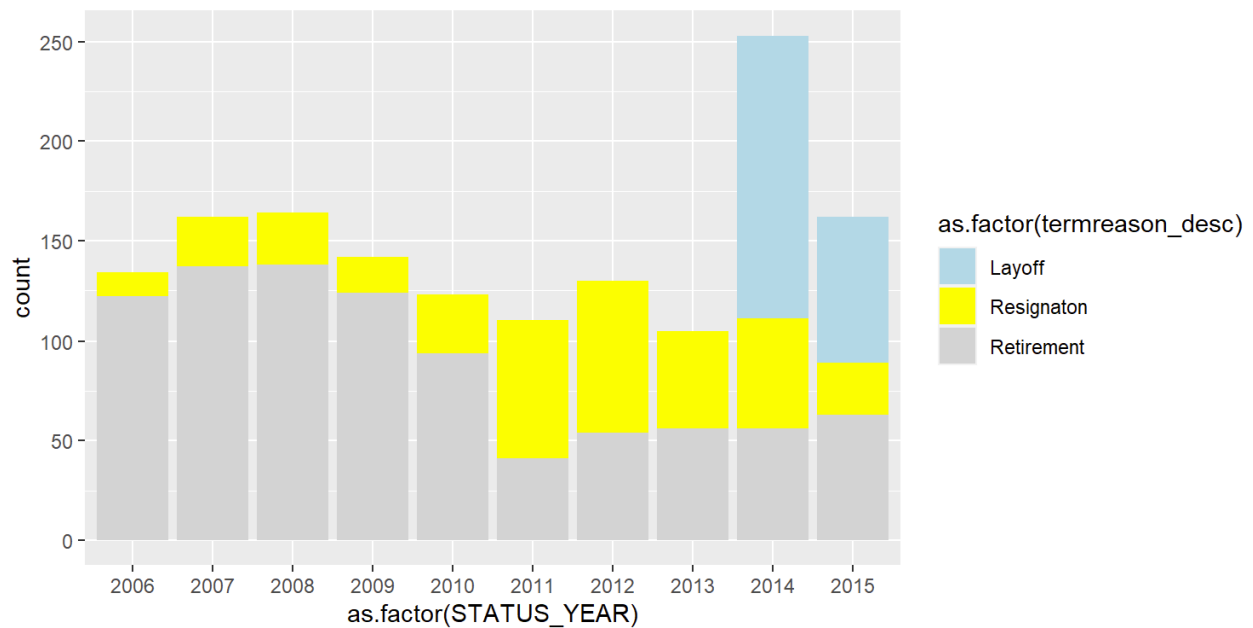
Just Terminates By Termination Type And Status Year

```
library(ggplot2)
ggplot() +
  geom_bar(
    aes(
      y = ..count..,
      x = as.factor(STATUS_YEAR),
      fill = as.factor(termtype_desc)
    ),
    data = TerminatesData,
    position = position_stack()
  ) +
  scale_fill_manual(
    values = c("Voluntary" = "lightblue", "Involuntary" = "yellow"))
```



Just Terminates By Status Year and Termination Reason

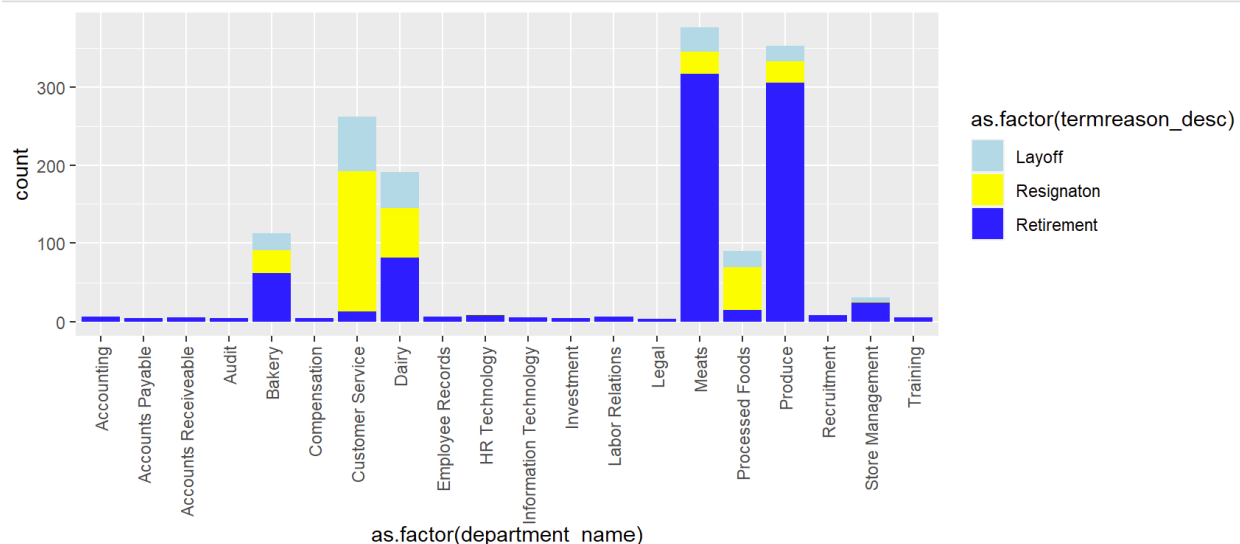
```
library(ggplot2)
ggplot() +
  geom_bar(
    aes(
      y = ..count..,
      x = as.factor(STATUS_YEAR),
      fill = as.factor(termreason_desc)
    ),
    data = TerminatesData,
    position = position_stack()
  ) +
  scale_fill_manual(
    values = c("Layoff" = "lightblue", "Resignaton" = "yellow", "Retirement"
  = "lightgray")
  )
```



It seems that there were layoffs in 2014 and 2015 which accounts for the involuntary terminations.

Just Terminates By Termination Reason and Department

```
library(ggplot2)
ggplot() +
  geom_bar(
    aes(
      y = ..count..,
      x = as.factor(department_name),
      fill = as.factor(termreason_desc)
    ),
    data = TerminatesData,
    position = position_stack()
  ) +
  scale_fill_manual(
    values = c("Layoff" = "lightblue", "Resignation" = "yellow", "Retirement" = "blue", "Unknown" = "gray")
  ) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```



When we look at the terminate by Department, a few things stick out. Customer Service has a much larger proportion of resignation compared to other departments. And retirement, in general, is high in a number of departments.

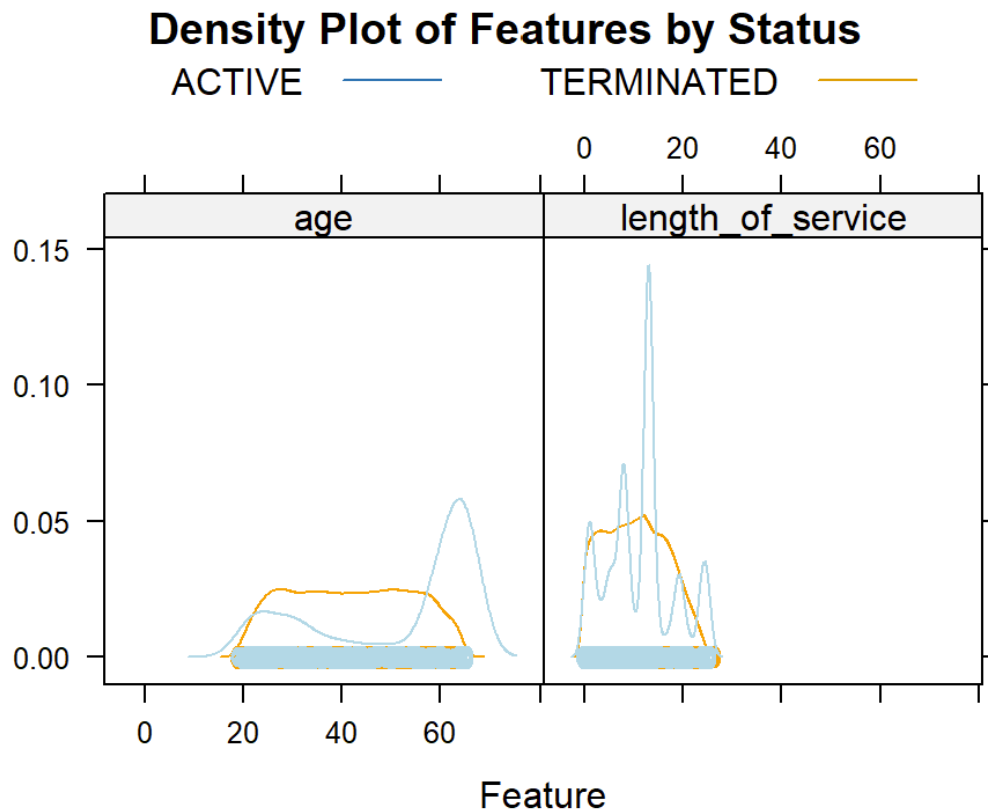
How does Age and Length of Service affect termination?

```
library(caret)

# specify the features (columns 6 and 7) and the response variable (STATUS)
features <- MYdataset[, 6:7]
response_variable <- as.factor(MYdataset$STATUS)

# Create a density plot using featurePlot
featurePlot(
  x = features,
  y = response_variable,
  plot = "density",
  auto.key = list(columns = 2),
  main = "Density Plot of Features by Status",
```

```
legend.title = "Status",
legend.labels = c("Active", "Terminated"),
col = c("orange", "lightblue"))
```



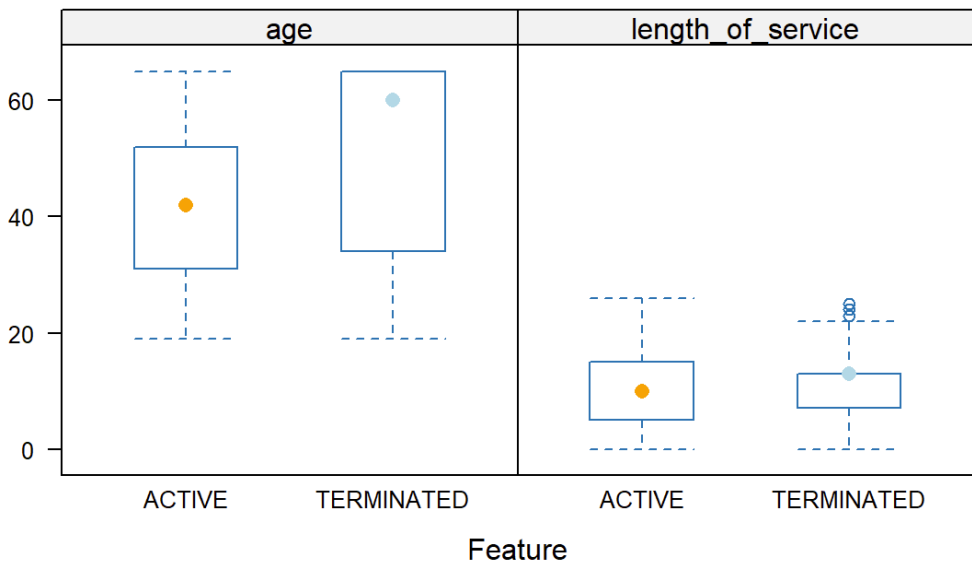
Density plots show some interesting things. For terminates, there is some elevation from 20 to 30 and a spike at 60. For length of service there are 5 spikes. One around 1 year, another one around 5 years, and a big one around 15 years, and a couple at 20 and 25 years.

Age and Length of Service Distributions By Status

```
library(caret)

featurePlot(
  x = MYdataset[, 6:7],
  y = as.factor(MYdataset$STATUS),
  plot = "box",
  auto.key = list(columns = 2),
  col = c("orange", "lightblue"), # Specify the colors for each status
  main = "Box Plot of Features by Status",
  legend.title = "Status",
  legend.labels = c("Active", "Terminated"))
```


Box Plot of Features by Status



A boxplot analysis shows a high average age for terminates as compared to active. Length of service shows not much difference between active and terminated.

Step 3 – Build The Model

It should be mentioned again that for building models, we never want to use **all** our data to build the model. This can lead to overfitting, where it might be able to predict well on current data that it sees as is built on, but may not predict well on data that it hasn't seen.

We have 10 years of historical data. This is a lot, usually, companies work only with a few years. In our case, we will use the first 9 to train the model, and the 10th year to test it. Moreover, we will use 10 fold cross-validation on the training data as well. So before we actually try out a variety of modeling algorithms, we need to partition the data into training and testing datasets.

Splitting The Data to Train and Test Data

```
library(rattle)
library(magrittr)
# Set a pre-defined value to reset the random seed for repeatability
set.seed(42)
# Load the dataset
MFG10YearTerminationData <- read.csv("C://Users/alire/Downloads/Datasets-Tutorial-People-Analytics/HRChurn.csv")
# Create training and testing datasets
MYnobs <- nrow(MFG10YearTerminationData)
MYsample <- subset(MFG10YearTerminationData, STATUS_YEAR <= 2014)
MYvalidate <- NULL
MYtest <- subset(MFG10YearTerminationData, STATUS_YEAR == 2015)

# variable selections
```

```

MYinput <- c("age", "length_of_service", "gender_full", "STATUS_YEAR", "BUSINESS_UNIT")
MYnumeric <- c("age", "length_of_service", "STATUS_YEAR")
MYcategoric <- c("gender_full", "BUSINESS_UNIT")
MYtarget <- "STATUS"
MYrisk <- NULL
MYident <- "EmployeeID"
MYignore <- c("recorddate_key", "birthdate_key", "orighiredate_key", "terminationdate_key", "city_name", "gender_short", "termreason_desc", "termtype_desc", "department_name", "job_title", "store_name")
MYweights <- NULL
# Create training and testing datasets with selected variables
MYTrainingData <- MYsample[c(MYinput, MYtarget)]
MYTestingData <- MYtest[c(MYinput, MYtarget)]

```

Decision Tree

```

library(rattle)
library(rpart, quietly=TRUE)

# Reset the random number seed to obtain the same results each time.
set.seed(crv$seed)

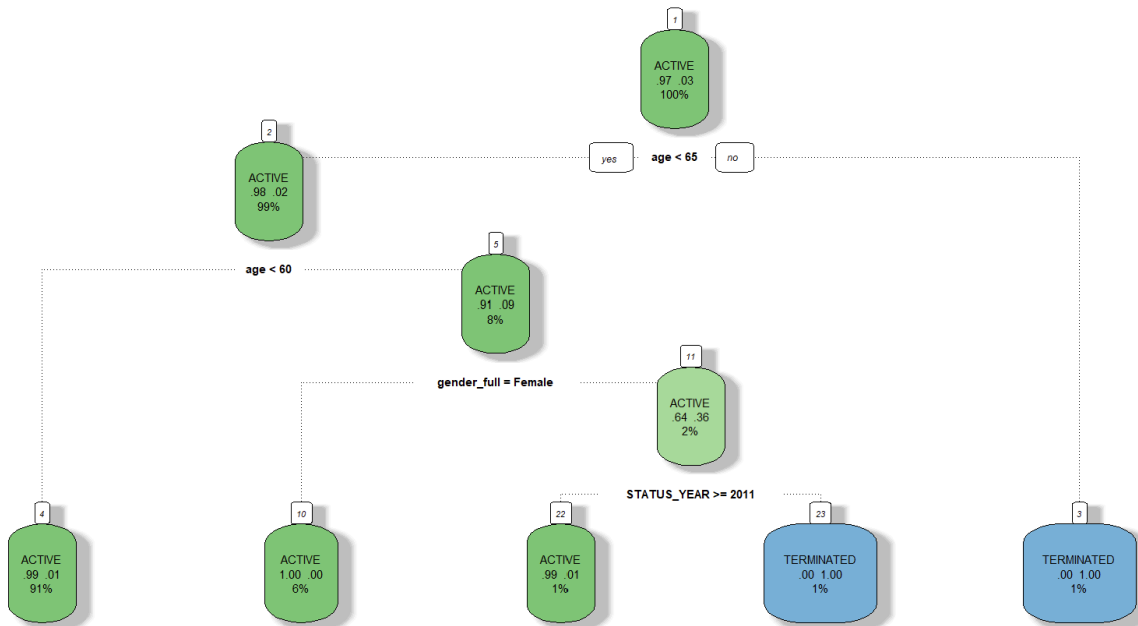
# Build the Decision Tree model.
MYrpart <- rpart(STATUS ~ .,
                 data = MYTrainingData,
                 method = "class",
                 parms = list(split = "information"),
                 control = rpart.control(usesurrogate = 0, maxsurrogate = 0))

# Generate a textual view of the Decision Tree model.
#print(MYrpart)
#printcp(MYrpart)
#cat("\n")

# Plot the resulting Decision Tree.
fancyRpartPlot(MYrpart, main = "Decision Tree MFG10YearTerminationData $ STATUS")

```

Decision Tree MFG10YearTerminationData \$ STATUS



```
# Rattle timestamp: 2016-03-25 18:50:22 x86_64-w64-mingw32
# Evaluate model performance.
# Generate an Error Matrix for the Decision Tree model.
# Obtain the response from the Decision Tree model.
```

```
MYpr <- predict(MYrpart, newdata = MYTestingData[c(MYinput, MYtarget)], type = "class")
```

```
# Generate the confusion matrix showing counts.
conf_matrix <- table(MYTestingData$STATUS, MYpr,
                     dnn = c("Actual", "Predicted"))
```

```
# Display the confusion matrix
conf_matrix
```

	Predicted	
Actual	ACTIVE	TERMINATED
ACTIVE	4799	0
TERMINATED	99	63

```
# Generate the confusion matrix showing proportions.
pcme <- function(actual, cl) {
  nc <- nrow(cl)
  tbl <- cbind(cl / length(actual),
               Error = sapply(1:nc,
                             function(r) round(sum(cl[r, -r]) / sum(cl[r, ]),
                             2)))
  names(attr(tbl, "dimnames")) <- c("Actual", "Predicted")
  return(tbl)
}
```

```
# Apply the pcme function to your actual and predicted values
per <- pcme(MYTestingData$STATUS, table(MYTestingData$STATUS, MYpr))
# Display the confusion matrix with proportions
round(per, 2)
```

Actual	Predicted		Error
	ACTIVE	TERMINATED	
ACTIVE	0.97	0.00	0.00
TERMINATED	0.02	0.01	0.61

```
# Calculate the overall error percentage.
> cat(100*round(1-sum(diag(per), na.rm=TRUE), 2))
2

# Calculate the averaged class error percentage.
> cat(100*round(mean(per[, "Error"], na.rm=TRUE), 2))
30
```

(Sundmark, n.d.)

References

Sundmark, L. (n.d.). *A Tutorial on People Analytics Using R – Employee Churn*. Retrieved from AIHR:
<https://www.aihr.com/blog/tutorial-people-analytics-r-employee-churn/>