CSDA1050 Capstone Project Sprint2

Eugene Yong Geun Park

2019-08-13

Background

Continuing from Sprint1, now predictive models are being built using refined dataset.

ML Modelling

Applying various ML techniques to create models which can provide insights. Starting with classification models as employees are to be classified as Active, Quit, and Terminated status. However, the focus is not just in classification. Other methods and models will be applied as necessary.

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#importing the original file. Feature engineering that was perfored in Sprint1 is rep
licated for Sprint2 modelling.
hr<-read.csv("~/Desktop/CSDA-1050F18S1/eugenepark/CSDA1050HR.csv")
hdate <- as.character(hr$Hire)
tdate <- as.character(hr$Termination)
bdate <- as.character(hr$DOB)
hr$hdate = as.Date(hdate, format="%Y%m%d")
hr$tdate = as.Date(tdate, format="%Y%m%d")
hr$bdate = as.Date(bdate, format="%Y%m%d")
hr$hireage <- as.integer(round((hr$hdate-hr$bdate)/365, digit=0))
hr$termage <- as.integer(round((hr$tdate-hr$bdate)/365, digit=0))
hr$current <-as.Date(Sys.Date())
hr$tenure <- ifelse(is.na(hr$termage), as.integer(round((hr$current-hr$hdate)/365, di
git=0))
                    ,as.integer(round((hr$tdate-hr$hdate)/365, digit=0)))
```

Decision Tree1:

Starting it off with a Decision Tree model. Prior to creating a model, I am subsetting the dataset into Train(80%) & Test(20%). For now, I am fitting in all variables that were explored in Sprint1.

```
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 3.5.2
```

```
#Employees' whose TermType is blank, filling them in as "Active" so modelling can go
smoothly.
levels(hr$TermType)[1] <-"Active"
hrmodel <- select(hr, C.LEVEL, Team, Job.Level, Team, Raise, Education, DistToWork, T
ermType, hireage, termage, tenure)
str(hrmodel)
```

```
## 'data.frame':    127 obs. of  10 variables:
##  $ C.LEVEL   : Factor w/ 6 levels "C1","C2","C3",..: 3 3 2 3 2 1 3 3 2 2 ...
##  $ Team      : Factor w/ 5 levels "Account Services",..: 3 3 1 1 5 1 4 5 3 3 ...
##  $ Job.Level : Factor w/ 5 levels "Associate","Manager",..: 5 5 5 5 1 1 5 5 1 1 ..
.
##  $ Raise     : Factor w/ 5 levels " -   ","1","2",..: 2 3 2 1 3 3 1 2 1 1 ...
##  $ Education : Factor w/ 4 levels "Bachelors Degree",..: 1 1 1 1 1 2 1 1 1 1 ...
##  $ DistToWork: num  15 40 25 20 13.3 24.5 23 45.3 8 5.2 ...
##  $ TermType  : Factor w/ 3 levels "Active","Quit",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ hireage   : int  32 41 27 34 26 28 28 39 32 29 ...
##  $ termage   : int  37 45 32 38 33 31 29 42 33 30 ...
##  $ tenure    : int  5 4 5 4 6 3 1 3 0 1 ...
```

```
set.seed(100)
sample <- sample.split(hrmodel, SplitRatio = 0.8)
train <- subset(hrmodel, sample==TRUE)
test <- subset(hrmodel, sample==FALSE)
```

```
prop.table((table(train$TermType)))
```

```
## 
##     Active       Quit Terminated 
##  0.1287129  0.6138614  0.2574257
```

```
prop.table((table(test$TermType)))
```

```
## 
##     Active       Quit Terminated 
##  0.1538462  0.6153846  0.2307692
```

Noting that this tree isn't classifying any of "Active" employees here.

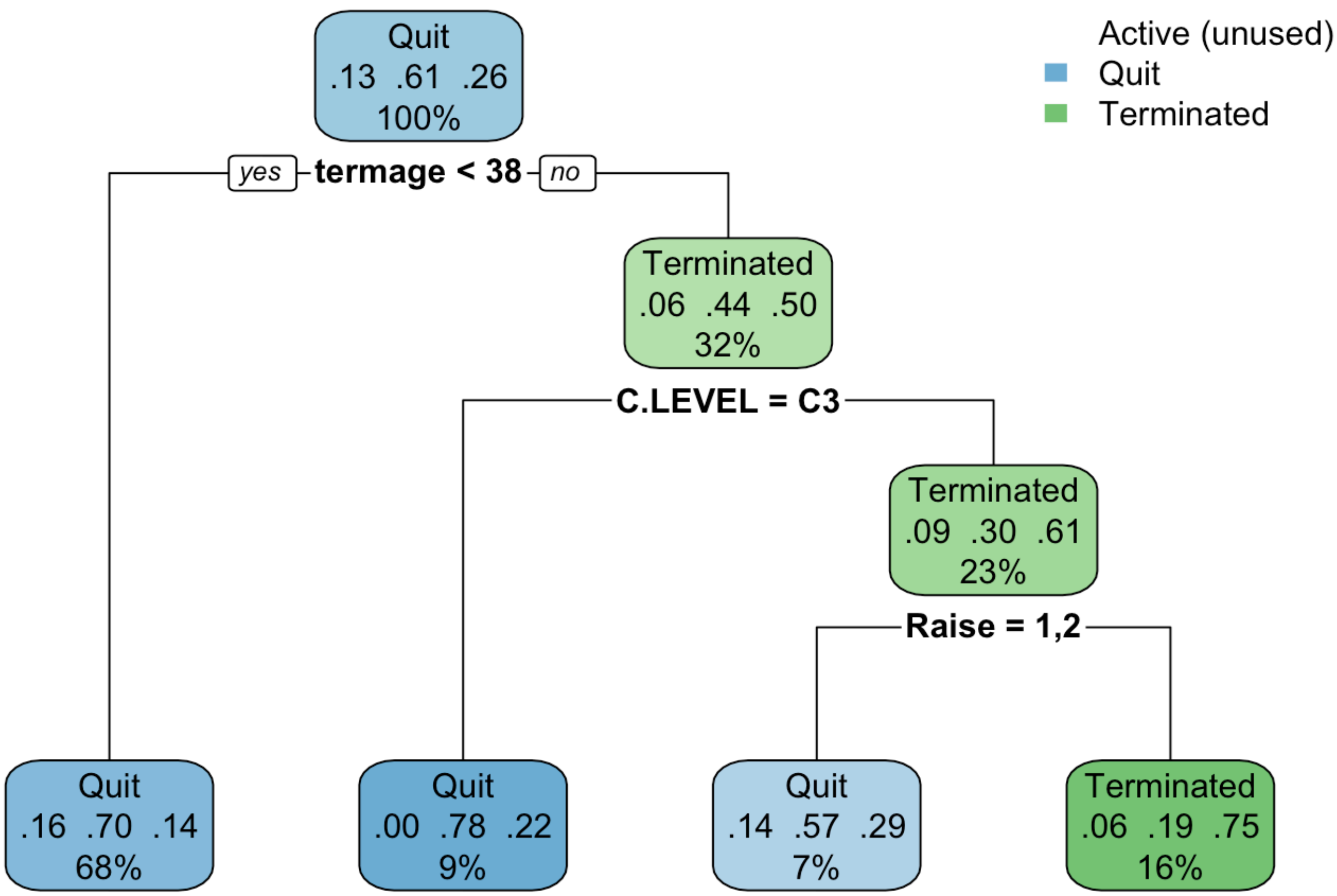
```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 3.5.2
```

```
library(tree)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 3.5.2
```

```
fit <- rpart(TermType~., data=train)
rpart.plot(fit)
```




```
summary(fit)
```

```
## Call:
## rpart(formula = TermType ~ ., data = train)
##   n= 101
##
##           CP nsplit rel error   xerror      xstd
## 1 0.08974359      0 1.0000000 1.000000 0.1254593
## 2 0.05128205      2 0.8205128 1.102564 0.1274157
## 3 0.01000000      3 0.7692308 1.025641 0.1260287
##
## Variable importance
```

```
##    termage    hireage    C.LEVEL      Raise  Job.Level DistToWork
##         26         24         22          9          8          5
##       Team     tenure
##          5          1
##
## Node number 1: 101 observations,    complexity param=0.08974359
##   predicted class=Quit        expected loss=0.3861386  P(node) =1
##     class counts:    13    62    26
##    probabilities: 0.129 0.614 0.257
##   left son=2 (69 obs) right son=3 (32 obs)
##   Primary splits:
##       termage   < 37.5  to the left,  improve=5.151306, (13 missing)
##       hireage   < 31.5  to the left,  improve=4.885526, (0 missing)
##       Job.Level splits as  LRRRL,     improve=4.404344, (0 missing)
##       C.LEVEL   splits as  LLLRRR,    improve=3.429813, (0 missing)
##       Education splits as  RLRR,      improve=2.807078, (0 missing)
##   Surrogate splits:
##       hireage    < 34.5  to the left,  agree=0.955, adj=0.867, (13 split)
##       C.LEVEL    splits as  LLLRRR,    agree=0.761, adj=0.300, (0 split)
##       Job.Level  splits as  LLRRL,     agree=0.761, adj=0.300, (0 split)
##       Team       splits as  LRLLR,     agree=0.727, adj=0.200, (0 split)
##       DistToWork < 28.45 to the left,  agree=0.693, adj=0.100, (0 split)
##
## Node number 2: 69 observations
##   predicted class=Quit        expected loss=0.3043478  P(node) =0.6831683
##     class counts:    11    48    10
##    probabilities: 0.159 0.696 0.145
##
## Node number 3: 32 observations,    complexity param=0.08974359
##   predicted class=Terminated  expected loss=0.5  P(node) =0.3168317
##     class counts:     2    14    16
##    probabilities: 0.062 0.438 0.500
##   left son=6 (9 obs) right son=7 (23 obs)
##   Primary splits:
##       C.LEVEL    splits as  RRLRRR,    improve=2.4649760, (0 missing)
##       Raise      splits as  RLLRR,     improve=2.2166670, (0 missing)
##       DistToWork < 32.25 to the right, improve=1.8871430, (0 missing)
##       Job.Level  splits as  RLRRL,     improve=1.7500000, (0 missing)
##       Education  splits as  LRR-,      improve=0.9577295, (0 missing)
##   Surrogate splits:
##       DistToWork < 30.3  to the right, agree=0.781, adj=0.222, (0 split)
##
## Node number 6: 9 observations
##   predicted class=Quit        expected loss=0.2222222  P(node) =0.08910891
##     class counts:     0     7     2
##    probabilities: 0.000 0.778 0.222
##
```

```
## Node number 7: 23 observations,    complexity param=0.05128205
##   predicted class=Terminated  expected loss=0.3913043  P(node) =0.2277228
##     class counts:     2     7    14
##    probabilities: 0.087 0.304 0.609
##   left son=14 (7 obs) right son=15 (16 obs)
##   Primary splits:
##       Raise      splits as  RLLRR,     improve=1.7989130, (0 missing)
##       DistToWork < 5.6   to the left,  improve=1.1572460, (0 missing)
##       tenure     < 2.5   to the right, improve=0.6977226, (0 missing)
##       C.LEVEL    splits as  RL-RLR,    improve=0.6572464, (0 missing)
##       Team       splits as  LLRRL,     improve=0.2453416, (0 missing)
##   Surrogate splits:
##       C.LEVEL splits as  LR-RRR,    agree=0.739, adj=0.143, (0 split)
##       hireage < 35.5  to the left,  agree=0.739, adj=0.143, (0 split)
##       tenure  < 2.5   to the right, agree=0.739, adj=0.143, (0 split)
##
## Node number 14: 7 observations
##   predicted class=Quit        expected loss=0.4285714  P(node) =0.06930693
##     class counts:     1     4     2
##    probabilities: 0.143 0.571 0.286
##
## Node number 15: 16 observations
##   predicted class=Terminated  expected loss=0.25  P(node) =0.1584158
##     class counts:     1     3    12
##    probabilities: 0.062 0.188 0.750
```

Checking the overall accuracy of the model. 53% overall accuracy is not reliable.

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.5.2
```

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

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
predict <- predict(fit, test, type='class')
confusionMatrix(predict, test$TermType)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Active Quit Terminated
##   Active          0    0          0
##   Quit            3   14          6
##   Terminated      1    2          0
##
## Overall Statistics
##
##                Accuracy : 0.5385
##                  95% CI : (0.3337, 0.7341)
##     No Information Rate : 0.6154
##     P-Value [Acc > NIR] : 0.8432
##
##                   Kappa : -0.0759
##
##  Mcnemar's Test P-Value : 0.1116
##
## Statistics by Class:
##
##                      Class: Active Class: Quit Class: Terminated
## Sensitivity                 0.0000      0.8750            0.0000
## Specificity                 1.0000      0.1000            0.8500
## Pos Pred Value                 NaN      0.6087            0.0000
## Neg Pred Value              0.8462      0.3333            0.7391
## Prevalence                  0.1538      0.6154            0.2308
## Detection Rate              0.0000      0.5385            0.0000
## Detection Prevalence        0.0000      0.8846            0.1154
## Balanced Accuracy           0.5000      0.4875            0.4250
```

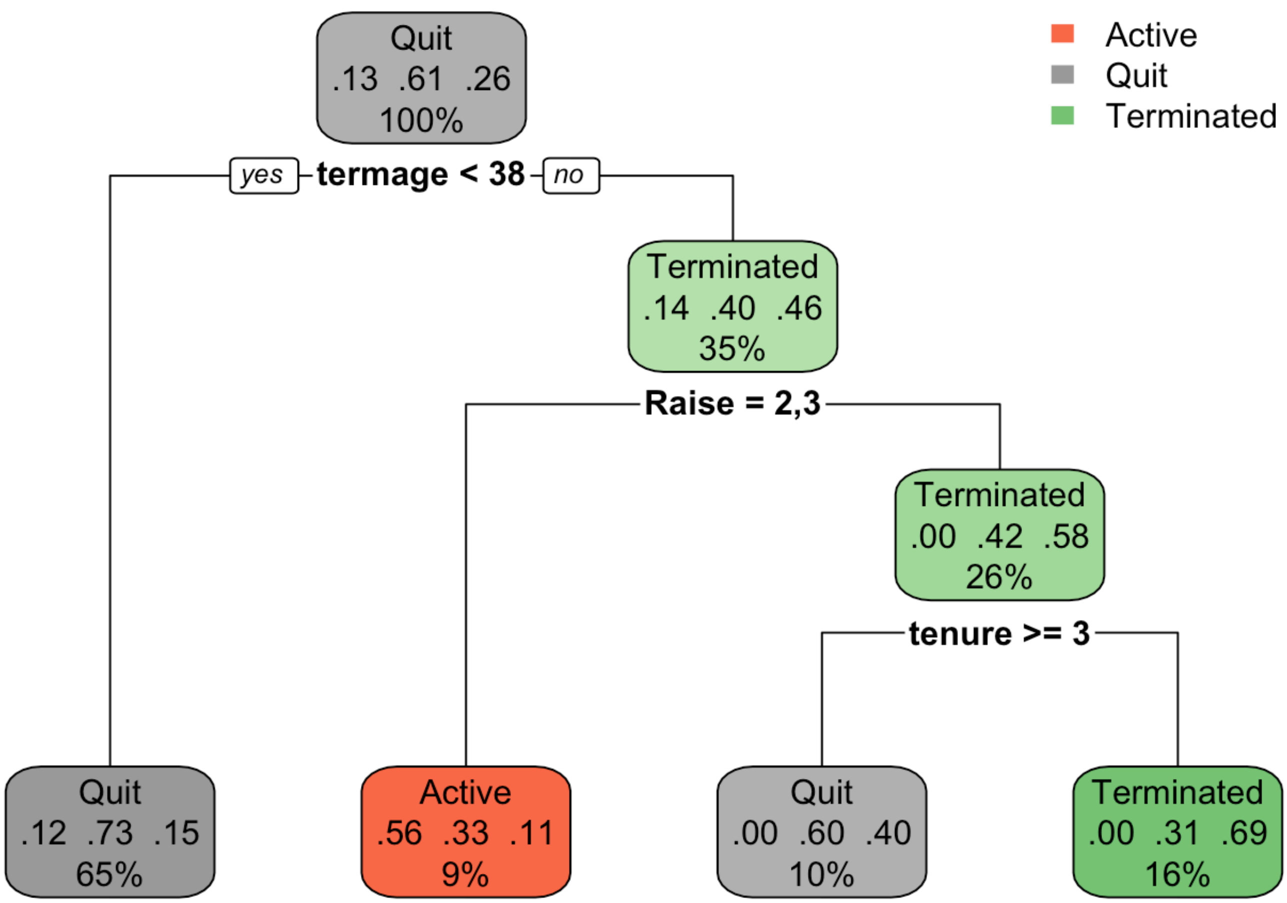
Decision Tree2:

Using Sprint1's insight, starting to explore variables that seemed more relevant than previous attempt.

As a result, visually, this tree has been able to classify all three statuses of employees: "Active", "Quit", and "Terminated", while previous model failed to.

However, overall accuracy and reliability have worsened.

```
#Chosen variables are tenure, Raise and termage.
hrmodel2 <- select(hr, TermType, tenure, Raise, termage)
set.seed(102)
sample2 <- sample.split(hrmodel2, SplitRatio = 0.8)
train2 <- subset(hrmodel2, sample==TRUE)
test2 <- subset(hrmodel2, sample==FALSE)
fit2 <- rpart(TermType~.,data=train2)
rpart.plot(fit2)
```




```
predict2 <- predict(fit2, test2, type='class')
confusionMatrix(predict2, test2$TermType)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Active Quit Terminated
##   Active          0    0          0
##   Quit            4   12          6
##   Terminated      0    4          0
##
## Overall Statistics
##
##                Accuracy : 0.4615
##                  95% CI : (0.2659, 0.6663)
##     No Information Rate : 0.6154
##     P-Value [Acc > NIR] : 0.9635
##
##                   Kappa : -0.2133
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Active Class: Quit Class: Terminated
## Sensitivity                 0.0000      0.7500            0.0000
## Specificity                 1.0000      0.0000            0.8000
## Pos Pred Value                 NaN      0.5455            0.0000
## Neg Pred Value              0.8462      0.0000            0.7273
## Prevalence                  0.1538      0.6154            0.2308
## Detection Rate              0.0000      0.4615            0.0000
## Detection Prevalence        0.0000      0.8462            0.1538
## Balanced Accuracy           0.5000      0.3750            0.4000
```

Decision Tree3:

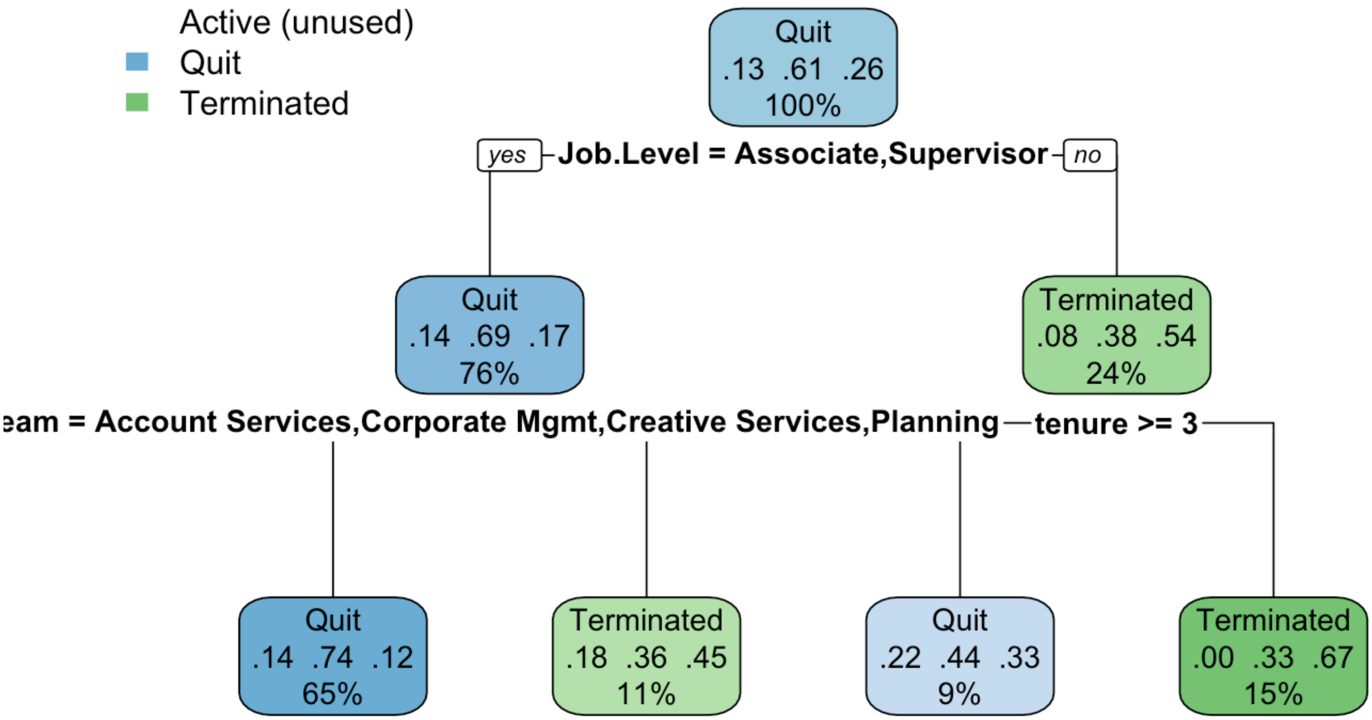
After numerous attempts of combining what seemed to be significant factors from Data Exploration, below tree is a model that gives higher accuracy (57%) than previous ones. Again, this can not be considered as a reliable model but it shares some insight.

Associate & Supervisor are relatively junior positions and the organization is not doing a good job retaining them as majority of them leaves voluntarily. On the other hand, employees who are not “associate” or “supervisor”" can be considered as more senior positions and they face more frequent cases of termination.

When drilled down deeper, among Associates & Supervisors, employees in Account, Corporate, Creative and Planning departments, tend to leave more voluntarily, which leaves Production department that has less portion of quitting. This makes sense because first mentioned 4 divisions are more Advertising specific roles which are sought after and more actively recruited. Production department’s role and scope do not tend to change from one company to another, which leads to less active recruitment.

And for senior positions, the tree indicates that tenure is one of significant factors, which also is logical. In company’s perspective, Senior position is a bigger investment. Their value and/or ROI is more closely monitored and retention decisions will have to be made timely for financial reasons.

```
hrmodel3 <- select(hr, TermType, tenure, Team, Job.Level)
set.seed(104)
sample3 <- sample.split(hrmodel3, SplitRatio = 0.8)
train3 <- subset(hrmodel3, sample==TRUE)
test3 <- subset(hrmodel3, sample==FALSE)
fit3 <- rpart(TermType~.,data=train3)
rpart.plot(fit3)
```




```
predict3 <- predict(fit3, test3, type='class')
confusionMatrix(predict3, test3$TermType)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Active Quit Terminated
##   Active          0    0          0
##   Quit            4   13          4
##   Terminated      0    3          2
##
## Overall Statistics
##
##                Accuracy : 0.5769
##                  95% CI : (0.3692, 0.7665)
##     No Information Rate : 0.6154
##     P-Value [Acc > NIR] : 0.7302
##
##                   Kappa : 0.0774
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Active Class: Quit Class: Terminated
## Sensitivity                 0.0000      0.8125           0.33333
## Specificity                 1.0000      0.2000           0.85000
## Pos Pred Value                 NaN      0.6190           0.40000
## Neg Pred Value              0.8462      0.4000           0.80952
## Prevalence                  0.1538      0.6154           0.23077
## Detection Rate              0.0000      0.5000           0.07692
## Detection Prevalence        0.0000      0.8077           0.19231
## Balanced Accuracy           0.5000      0.5062           0.59167
```

Random Forest1:

Trying a Random Forest model, using the same dataset(variables) from Decision Tree 3 as it had given the highest accuracy so far. Unfortunately, the result does not seem to provide any improvement in accuracy.

```
library(randomForest)
```

```
## randomForest 4.6-14
```

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

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

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

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
set.seed(47)
rfmodel3 <- randomForest(TermType ~., data=train3, proximity=TRUE)
rfmodel3
```

```
##
## Call:
##  randomForest(formula = TermType ~ ., data = train3, proximity = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 40.59%
## Confusion matrix:
##            Active Quit Terminated class.error
## Active          0   11          2   1.0000000
## Quit            1   55          6   0.1129032
## Terminated      0   21          5   0.8076923
```

```
rfpredict3 <- predict(rfmodel3, test3, type='class')
confusionMatrix(rfpredict3, test3$TermType)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Active Quit Terminated
##   Active          0    1          0
##   Quit            4   14          5
##   Terminated      0    1          1
##
## Overall Statistics
##
##                Accuracy : 0.5769
##                  95% CI : (0.3692, 0.7665)
##     No Information Rate : 0.6154
##     P-Value [Acc > NIR] : 0.7302
##
##                   Kappa : 0.0205
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Active Class: Quit Class: Terminated
## Sensitivity                0.00000      0.8750           0.16667
## Specificity                0.95455      0.1000           0.95000
## Pos Pred Value             0.00000      0.6087           0.50000
## Neg Pred Value             0.84000      0.3333           0.79167
## Prevalence                 0.15385      0.6154           0.23077
## Detection Rate             0.00000      0.5385           0.03846
## Detection Prevalence       0.03846      0.8846           0.07692
## Balanced Accuracy          0.47727      0.4875           0.55833
```

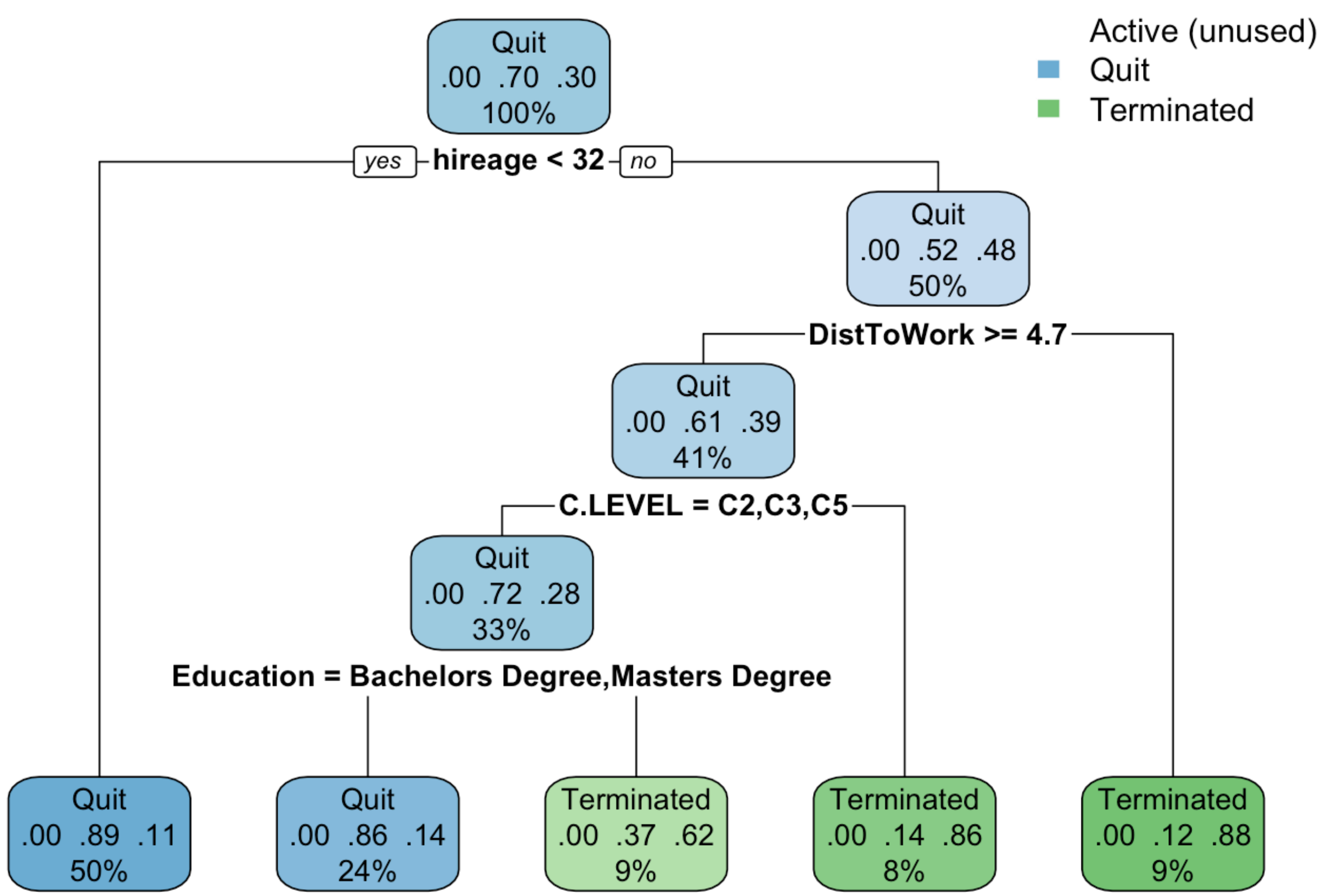
Decision Tree4:

Although I start to realize that size of my dataset sets limit to building a reliable model, I start to wonder if classifying “Active” employee is actually adding value to this analysis. Perhaps, the analysis should focus on characterstics of “Quit” employees and “Terminated” employees. Then the discovery can be applied to “Active” employees for operational action plan. In addition, removing a classification with smallest data

size and least accurary (Again, “Active” status) from confusion matrix might reveal how this model can truly perform classifying “Quit” and “Terminated” status.

This is the same dataset used in very first Decision Tree1 and by excluding “Active” status from the dataset, the accuracy has improved from 53% to 63%.

```
newhrmodel<-hrmodel[!(hrmodel$TermType=="Active"),]
set.seed(111)
newsample <- sample.split(newhrmodel, SplitRatio = 0.8)
newtrain <- subset(newhrmodel, sample==TRUE)
newtest <- subset(newhrmodel, sample==FALSE)
newfit <- rpart(TermType~.,data=newtrain)
rpart.plot(newfit)
```



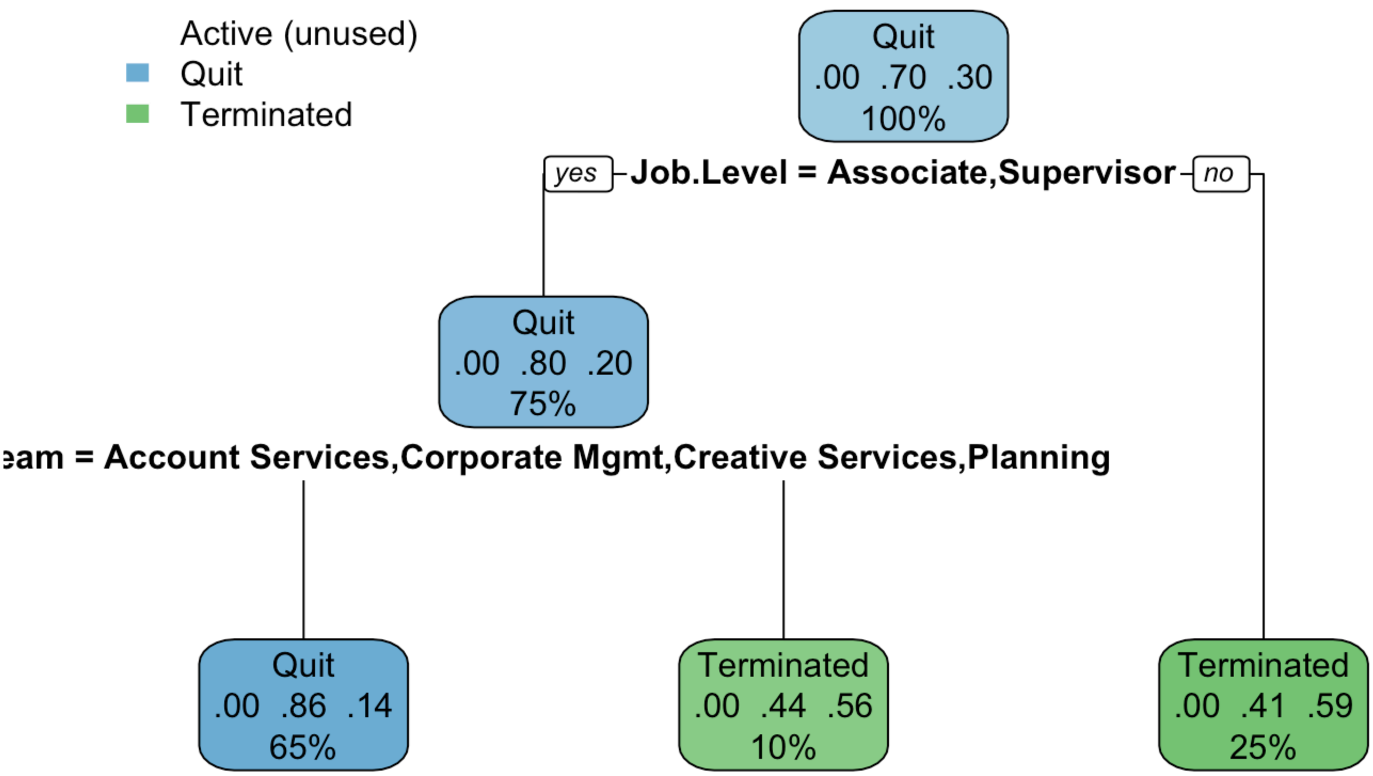
```
newpredict <- predict(newfit, newtest, type='class')
confusionMatrix(newpredict, newtest$TermType)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Active Quit Terminated
##   Active          0    0          0
##   Quit            0   12          4
##   Terminated      0    4          2
##
## Overall Statistics
##
##                Accuracy : 0.6364
##                  95% CI : (0.4066, 0.828)
##     No Information Rate : 0.7273
##     P-Value [Acc > NIR] : 0.8822
##
##                   Kappa : 0.0833
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Active Class: Quit Class: Terminated
## Sensitivity                     NA      0.7500           0.33333
## Specificity                      1      0.3333           0.75000
## Pos Pred Value                  NA      0.7500           0.33333
## Neg Pred Value                  NA      0.3333           0.75000
## Prevalence                       0      0.7273           0.27273
## Detection Rate                   0      0.5455           0.09091
## Detection Prevalence             0      0.7273           0.27273
## Balanced Accuracy               NA      0.5417           0.54167
```

Decision Tree5:

This now tests the Decision tree3 which had 57% accuracy. And excluding “Active” status has improved the model to 68%.

```
newhrmodel3<-hrmodel3[!(hrmodel3$TermType=="Active"),]
set.seed(113)
newsample3 <- sample.split(newhrmodel3, SplitRatio = 0.8)
newtrain3 <- subset(newhrmodel3, sample==TRUE)
newtest3 <- subset(newhrmodel3, sample==FALSE)
newfit3 <- rpart(TermType~.,data=newtrain3)
rpart.plot(newfit3)
```




```
newpredict3 <- predict(newfit3, newtest3, type='class')
confusionMatrix(newpredict3, newtest3$TermType)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Active Quit Terminated
##   Active          0    0          0
##   Quit            0   12          3
##   Terminated      0    4          3
##
## Overall Statistics
##
##                Accuracy : 0.6818
##                  95% CI : (0.4513, 0.8614)
##     No Information Rate : 0.7273
##     P-Value [Acc > NIR] : 0.7689
##
##                   Kappa : 0.2376
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Active Class: Quit Class: Terminated
## Sensitivity                     NA      0.7500            0.5000
## Specificity                      1      0.5000            0.7500
## Pos Pred Value                  NA      0.8000            0.4286
## Neg Pred Value                  NA      0.4286            0.8000
## Prevalence                       0      0.7273            0.2727
## Detection Rate                   0      0.5455            0.1364
## Detection Prevalence             0      0.6818            0.3182
## Balanced Accuracy               NA      0.6250            0.6250
```

Although there is a small improvement in Decision Tree model, it is concluded that a reliable classification model can not be built based on this dataset. I have decided to look into correlation between some hiring data and annual revenue, using linear regression.

```
#When new hires are made using recruiting firm, 22% of new hire's salary is paid as c
ommission. I am making a new column that illustrates commission paid to recruting fir
m for each hire.
hr$hirecost <- hr$BEGIN.SALARY * 0.22
#Now by aggregating hirecost by year, I get aggregated yearly total.
hirecost <- hr %>% group_by(H.Year) %>% summarise_each(funs(sum), hirecost)
```

```
## Warning: funs() is soft deprecated as of dplyr 0.8.0
## please use list() instead
##
##   # Before:
##   funs(name = f(.))
##
##   # After:
##   list(name = ~ f(.))
## This warning is displayed once per session.
```

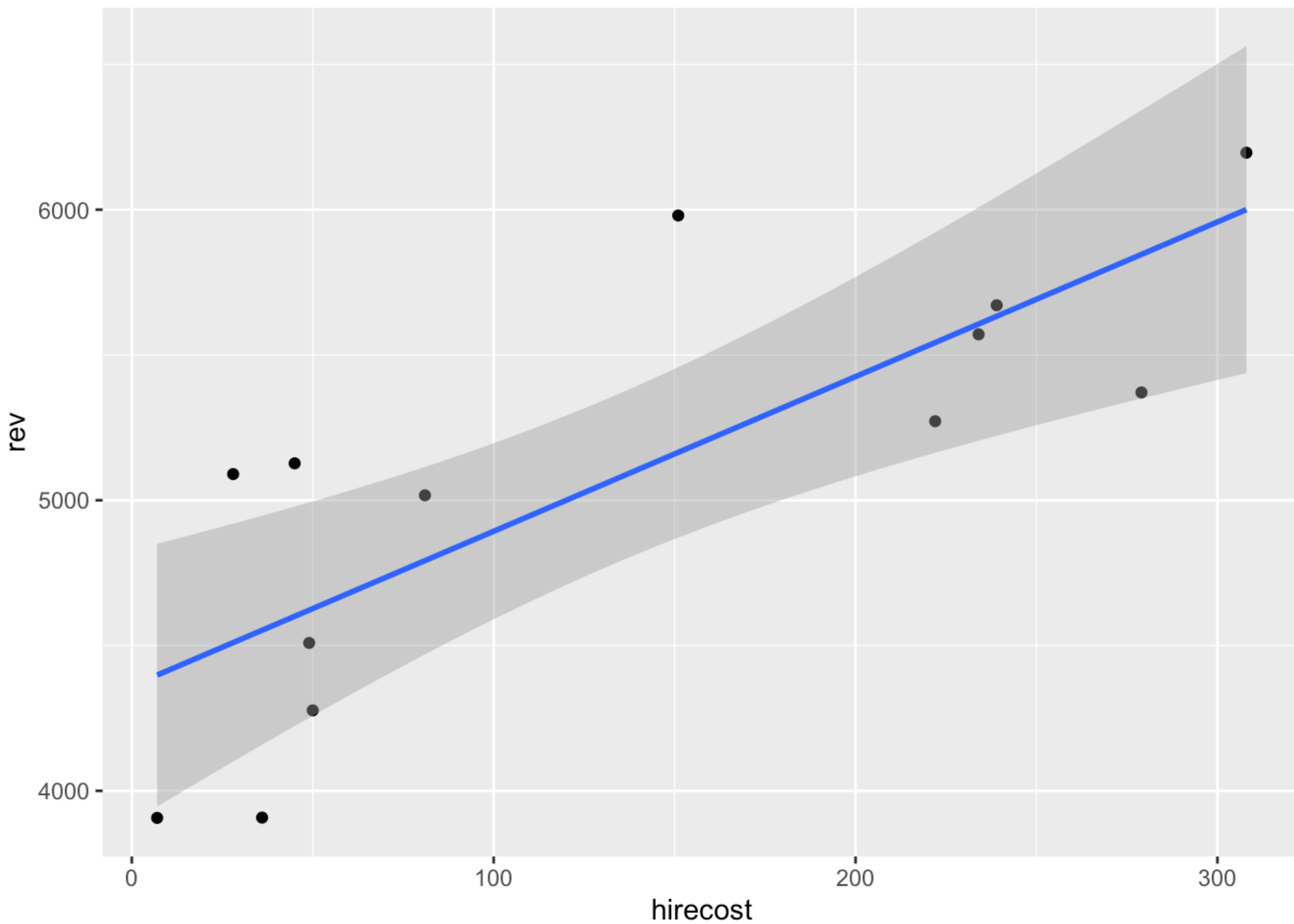
```
#An aggregation of total count of hires by year.
hirecount <- hr %>% count(H.Year)
#hirecost and hirecount are now being combined into a new dataset.
colnames(hirecost) <- c("year", "hirecost")
colnames(hirecount) <- c("year", "hirecount")
hire <- merge(hirecost, hirecount, KEY="year")
#Excluding 2019 row as the data is still subject to change.
hire <- hire[-c(10),]
#Adding revenue data manually for each year.
hire$rev <- c(3908146, 4509822, 3907264, 4277165, 5127230, 5571537, 5371010, 5671345,
6196730, 5980433, 5272719, 5017569, 5090057)
#A small clean up of dataset. Reducting numeric figures to thousands, then adding ave
rage cost per hire column.
hire$hirecost <-as.integer(hire$hirecost/1000)
hire$avgcost <- as.integer(hire$hirecost/hire$hirecount)
hire$rev <-as.integer(hire$rev/1000)

hire
```

```
##    year hirecost hirecount  rev avgcost
## 1  2006       36         2 3908      18
## 2  2007       49         3 4509      16
## 3  2008        7         1 3907       7
## 4  2009       50         2 4277      25
## 5  2010       45         2 5127      22
## 6  2011      234        12 5571      19
## 7  2012      279        20 5371      13
## 8  2013      239        18 5671      13
## 9  2014      308        16 6196      19
## 11 2016      151        11 5980      13
## 12 2017      222        14 5272      15
## 13 2018       81         6 5017      13
## 14 2019       28         2 5090      14
```

Linear Regression1:

Plotting hirecost and revenue together to check its correlation visually. Although it is a very small # of observations, there's a correlation.

```
library(ggplot2)
ggplot(hire, aes(x=hirecost, y=rev)) + geom_point() +geom_smooth(method='lm')+ scale_
x_continuous(labels = scales::comma)
```



Running a linear regression model to find out its coefficients.

```
lm1 <- lm(hirecost ~ rev, data = hire)
summary(lm1)
```

```
##
## Call:
## lm(formula = hirecost ~ rev, data = hire)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -107.459  -45.942    9.566   42.421  110.755
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -458.41206  140.66161  -3.259  0.00761 **
## rev            0.11667    0.02748   4.245  0.00138 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 70.03 on 11 degrees of freedom
## Multiple R-squared:  0.621,  Adjusted R-squared:  0.5865
## F-statistic: 18.02 on 1 and 11 DF,  p-value: 0.001377
```

From 2018’s 5090K revenue figure, if the organization is targetting 500K revenue growth year-over-year, below are hiring costs predicted using current model. 183K, 241K, 300K, & 358K in order.

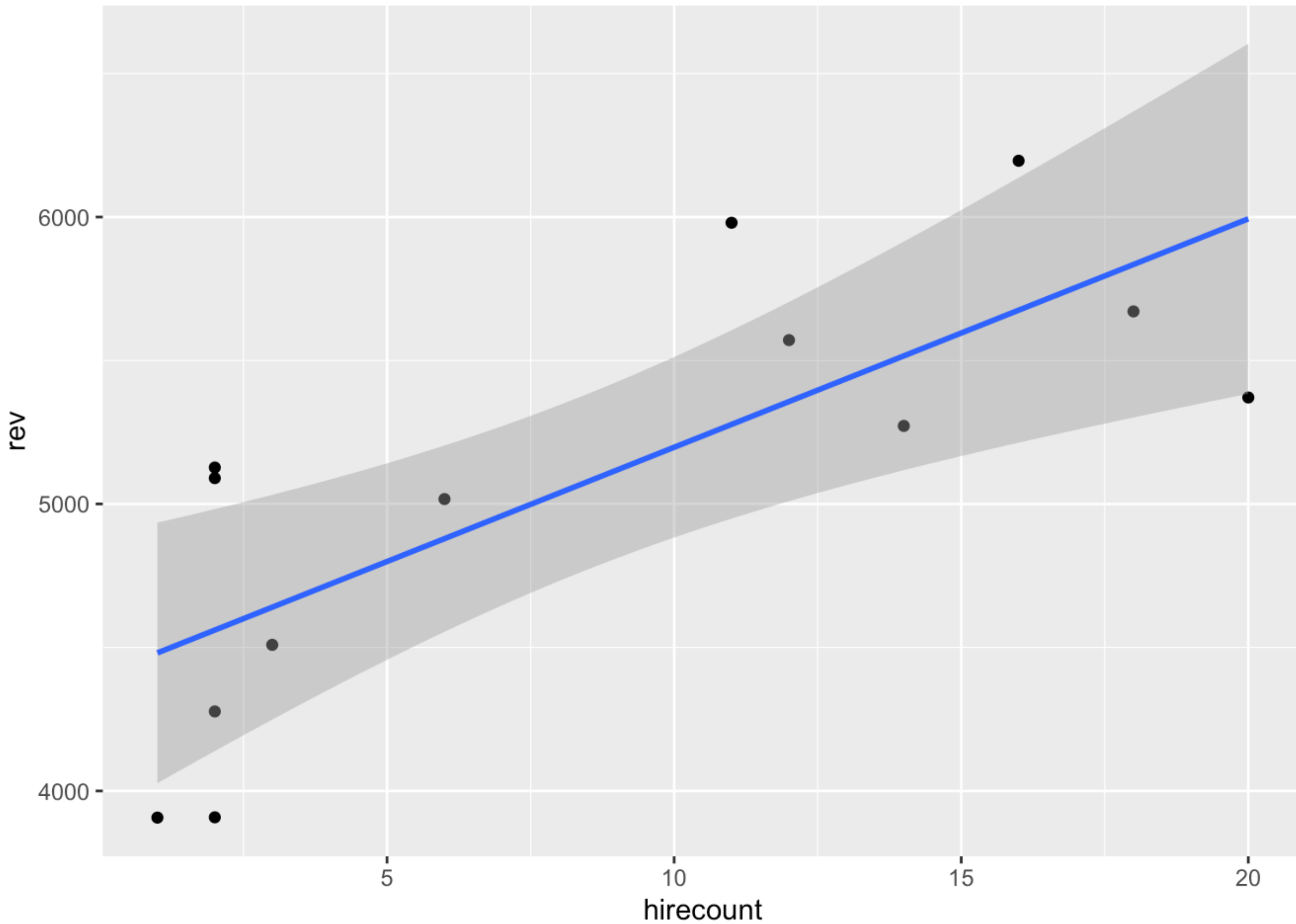
```
predict(lm1, newdata = data.frame((rev = c(5500, 6000, 6500, 7000))))
```

```
##        1        2        3        4
## 183.2955 241.6326 299.9696 358.3067
```

Linear Regression2:

Now plotting # of hires and revenue together.

```
ggplot(hire, aes(x=hirecount, y=rev)) + geom_point() +geom_smooth(method='lm')+ scale
_x_continuous(labels = scales::comma)
```



A linear model between # of hires & revenue. A correlation can be observed again.

```
lm2 <- lm(hirecount ~ rev, data = hire)
summary(lm2)
```

```
##
## Call:
## lm(formula = hirecount ~ rev, data = hire)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.8042 -2.0095 -0.5267  2.0019  9.4332
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -28.233459   9.580587  -2.947  0.01328 *
## rev           0.007224   0.001872   3.859  0.00266 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.77 on 11 degrees of freedom
## Multiple R-squared:  0.5752, Adjusted R-squared:  0.5365
## F-statistic: 14.89 on 1 and 11 DF,  p-value: 0.002658
```

When the same 500K year-over-year revenue growth is applied, the model suggests below # of new hires prediction.

```
predict(lm2, newdata = data.frame((rev = c(5500, 6000, 6500, 7000))))
```

```
##        1        2        3        4
## 11.49873 15.11075 18.72276 22.33478
```

Overall, this linear model analysis suggests that predicted hiring cost can be considered as opportunity cost to retain existing staff. The amounts are not insignificant. I feel that they are meaningful enough to actually propose solid career advancement & growth opportunity to individuals at risk, in a form of education and/or training.

```
predict(lm1, newdata = data.frame((rev = c(5500, 6000, 6500, 7000))))
```

```
##        1        2        3        4
## 183.2955 241.6326 299.9696 358.3067
```

```
predict(lm2, newdata = data.frame((rev = c(5500, 6000, 6500, 7000))))
```

```
##        1        2        3        4
## 11.49873 15.11075 18.72276 22.33478
```

Conclusion

Sprint2 analysis has started off with classification models such as Decision Tree and Random Forest. Although I was successful in increasing accuracy slightly, the overall performance was not satisfactory. Perhaps the failure was foreseen, given the limitation from small dataset. This does not mean that this analysis did not share any insights. As it was shown Data Exploration phase, this modelling has illustrated that more junior staff chooses to quit and more senior staff are terminated from the organization. Also bigger percentage in termination of senior staff hints that the organization tends to scrutinize senior staff's performance. Another discovery is that there is correlation between revenue and hiring cost. While it seemed obvious that hiring cost increase as revenue increases, it is very meaningful to quantify hiring cost in different business situations so it can help build budget for employee retention purpose.

Next step

Discoveries from sprint 1 and 2 will be put into a report for stakeholders. The goal is to create a report that insightful and beneficial to related party who are not necessarily familiar with machine learning.