Predicting Employee Turnover

Aaron Johnson

2/27/2021

Problem Statement

A data set has been provided by a company. We are interested in creating a model to predict which employees are most likely to leave the company.

Cleaning Employee Data

The data for this exercise was provided in 3 excel sheet: *'employee_data.xlsx'*, *'employee_reviews.xlsx'* and *'Employee_survey.xlsx'*
The process to clean the data was to remove *NULL* values into appropriate and combine the separate files into a single dataframe, which is the format most machine learning algorithms in R require.

Processing employee_data

```
library(tidyverse)
library(ggplot2)
library(caret)
library(readxl)
library(lubridate)
library(randomForest)
library(ROSE)

rm(list = ls())

setwd('C:/Users/aaron/OneDrive/Documents/Slalom Data Project')
employee_data <- read_excel('employee_data.xlsx')
employee_review <- read_excel('employee_reviews.xlsx')
employee_survey <- read_excel('Employee_survey.xlsx')

employee_data <- employee_data %>% replace_na(
  list(DegreeField ='None',MaritalStatus = 'Unknown', NumPreviousCompanies = 0,
      EmploymentEndReason ='0')
  ) %>%
  mutate(DegreeCompleted = factor(DegreeCompleted)) %>%
  mutate(DegreeField = factor(DegreeField)) %>%
  mutate(Department = as.factor(Department)) %>%
  mutate(Gender = as.factor(Gender)) %>%
  mutate(JobLevel = as.factor(JobLevel)) %>%
  mutate(MaritalStatus = as.factor(MaritalStatus)) %>%
  mutate(TravelFrequency = str_to_title(TravelFrequency)) %>%
  mutate(TravelFrequency = as.factor(TravelFrequency)) %>%
```

```
  mutate(EmploymentEndReason = if_else(EmploymentEndReason == 'Went to another company','1','0'))  %>%
  mutate(EmploymentEndReason = as.factor(EmploymentEndReason)) %>%
  mutate(TrainingsAttended = if_else(TrainingsAttended < 0, 0, TrainingsAttended)) %>%
  mutate(EmploymentStartDate = as.numeric(floor((date('2019-12-31')-date(EmploymentStartDate))/365.25))
  rename(YearsAtCompany = EmploymentStartDate) %>%
  mutate(YearOfBirth = 2019 - YearOfBirth) %>%
  rename(Age = YearOfBirth)
```

The general employee data required the most cleaning. 4 fields had NA's. Marital Status and DegreeField had new categories to capture the NA's while for Previous Companies worked and Employee end date it was assumed NA's represented 0 and still employed.Years at company and age were calculated as features from employee start date and year of birth. TrainingsAttended had at least one entry that was obviously wrong (-1) and this was coerced to 0. There were not any other errors that are evident from context, but data cleansing is an ongoing process and as more errors are found they should be corrected.

Processing employee_review

```
employee_review_sum <- employee_review %>% group_by(EmployeeId) %>%
  summarise(
    avg_review = mean(PerformanceRating),
    sd_review = sd(PerformanceRating),
    max_review = max(PerformanceRating),
    min_review = min(PerformanceRating)) %>%
  replace_na(list(sd_review = 0))

employee_review <- employee_review %>% group_by(EmployeeId) %>%
  arrange(desc(ReviewDate), .by_group = TRUE) %>%
  mutate(Review_num =row_number()) %>%
  mutate(Review_num = paste('Last',as.character(Review_num),'review',sep = '_')) %>%
  select(one_of(c('EmployeeId','Review_num','PerformanceRating'))) %>%
  pivot_wider(names_from = Review_num, values_from = PerformanceRating) %>%
  select(one_of('EmployeeId','Last_1_review'))

employee_review_sum <- inner_join(employee_review_sum,employee_review,by = "EmployeeId")
```

Employees had varying numbers of reviews, with up to 30 in some cases. To facilitate this data being incorporated into a model, the data had to be unpivoted so as there was a single record per employee. In addition, averages and standard deviation of reviews were calculated for employees. While there were many employees with more than 5 reviews, the vast majority has less than 3. To avoid having NA's for many employees, only the last review was kept. NA's could have been replaced with the average or median review but this may add a lot of unnecessary noise as most employees had only 1 or 2 reviews.

Processing employee_survey

```
employee_survey <- employee_survey %>% mutate(EmployeeId = str_replace(EmployeeId,'\\D','')) %>%
  mutate(EmployeeId = as.numeric(EmployeeId)) %>%
  mutate(Response =
           case_when(
             Response == 'Neither Satisfied nor Unsatisfied' ~ 3,
             Response == 'Somewhat Satisfied' ~ 4,
             Response == 'Somewhat Unsatisfied' ~ 2,
             Response == 'Very Satisfied' ~ 5,
             Response == 'Very Unsatisfied' ~ 1,
```

```
             Response == 'Very Poor' ~ 1,
             Response == 'Poor' ~ 2,
             Response == 'Fair' ~ 3,
             Response == 'Good' ~ 4,
             Response == 'Excellent' ~ 5
             )
         ) %>%
  mutate(Response = factor(Response,ordered = TRUE, levels = c(1,2,3,4,5))) %>%
  select(one_of('EmployeeId','QuestionNum','Response')) %>%
  pivot_wider(names_from = QuestionNum, values_from = Response)
```

The employee survey results were unpivoted so there was 1 record per employee. The responses had some clear order to them and were processed as such. There were some records with odd formatting for the employeeID field and this was corrected as appropriate.

Creating Training and Testing Set

```
data <- left_join(employee_data,employee_review_sum, by = "EmployeeId")
data <- left_join(data,employee_survey, by = "EmployeeId")

data <- data %>%
  replace_na(list(
    avg_review = 0,sd_review = 0, max_review = 0,min_review = 0,Last_1_review = 0)
    ) %>%
  mutate(Has_no_review = if_else(avg_review == 0, 1, 0)) %>%
  mutate(Has_no_review = as.factor(Has_no_review)) %>%
  select(!EmploymentEndDate) %>% select(!WeeklyHoursBudgeted) %>% select(!EmployeeId)

set.seed(2021)
dataDivide <- createDataPartition(data$EmploymentEndReason, p = .75, list = FALSE)

data_train <- data[dataDivide,]
data_test <- data[-dataDivide,]
X <- model.matrix(EmploymentEndReason~.,data = data,)[,-1]

Y <- data$EmploymentEndReason

Y_train <- Y[dataDivide]
X_train <- X[dataDivide,]
Y_test<- Y[-dataDivide]
X_test <- X[-dataDivide,]

summary(data)
```

```
##  CommuteDistance      DegreeCompleted          DegreeField
##  Min.   : 1.000   Associate    :  79   Other           :494
##  1st Qu.: 7.000   Bachelor     :1030   Business        :385
##  Median : 9.000   Below College:  23   Marketing       :364
##  Mean   : 9.587   Doctor       :  44   Finance         :312
##  3rd Qu.:12.000   Master       : 825   Computer Science:197
##  Max.   :25.000                        Life Sciences   :117
##                                        (Other)         :132
##                  Department  EmploymentEndReason YearsAtCompany
##  Accounting           :297   0:1604              Min.   : 0.000
```

```
##  Human Resources       : 96   1: 397             1st Qu.: 1.000
##  Information Technology:197                      Median : 3.000
##  Marketing             :414                      Mean   : 4.382
##  Other                 :817                      3rd Qu.: 6.000
##  Sales                 :180                      Max.   :38.000
##
##     Gender     JobLevel  MaritalStatus  NumPreviousCompanies NumYearsWorked
##  Female:1014   1:670    Divorced: 197   Min.   : 0.000       Min.   : 1.00
##  Male  : 987   2:645    Married :1007   1st Qu.: 1.000       1st Qu.: 7.00
##                3:439    Single  : 729   Median : 1.000       Median :14.00
##                4:198    Unknown :  68   Mean   : 1.791       Mean   :14.77
##                5: 49                    3rd Qu.: 3.000       3rd Qu.:21.00
##                                         Max.   :10.000       Max.   :49.00
##
##   OvertimeDays    OvertimeHours        Salary       TrainingsAttended
##  Min.   : 0.000   Min.   : 0.000   Min.   : 22700   Min.   :0.00
##  1st Qu.: 1.000   1st Qu.: 4.000   1st Qu.: 47100   1st Qu.:1.00
##  Median : 2.000   Median : 7.000   Median : 60300   Median :1.00
##  Mean   : 3.876   Mean   : 9.241   Mean   : 67543   Mean   :1.39
##  3rd Qu.: 5.000   3rd Qu.:12.000   3rd Qu.: 82200   3rd Qu.:2.00
##  Max.   :57.000   Max.   :78.000   Max.   :258300   Max.   :8.00
##
##          TravelFrequency      Age          avg_review      sd_review
##  Less Than Monthly:190    Min.   :24.00   Min.   :0.000   Min.   :0.0000
##  Monthly          :629    1st Qu.:31.00   1st Qu.:2.667   1st Qu.:0.0000
##  No Travel        :392    Median :37.00   Median :3.143   Median :0.5774
##  None             :391    Mean   :38.24   Mean   :2.689   Mean   :0.5334
##  Weekly           :399    3rd Qu.:45.00   3rd Qu.:3.500   3rd Qu.:0.8498
##                           Max.   :70.00   Max.   :5.000   Max.   :2.1213
##
##    max_review      min_review    Last_1_review   Q1      Q2      Q3
##  Min.   :0.000   Min.   :0.000   Min.   :0.000   1: 63   1: 64   1: 20
##  1st Qu.:3.000   1st Qu.:2.000   1st Qu.:2.000   2:226   2:199   2:164
##  Median :4.000   Median :2.000   Median :3.000   3:460   3:452   3:429
##  Mean   :3.293   Mean   :2.075   Mean   :2.717   4:589   4:641   4:789
##  3rd Qu.:4.000   3rd Qu.:3.000   3rd Qu.:4.000   5:663   5:645   5:599
##  Max.   :5.000   Max.   :5.000   Max.   :5.000
##
##  Q4      Has_no_review
##  1: 79   0:1632
##  2:198   1: 369
##  3:433
##  4:645
##  5:646
##
##
```

We can see a final summary of the model data. All looks well.

Modelling

Sample Test Model

```
test_model <- glm(EmploymentEndReason~., data = data, family = binomial)
test_pred <- predict(test_model,data,type = 'response')
test_pred <- ifelse(test_pred > .5,1,0)
(mean(data$EmploymentEndReason == test_pred))
```

```
## [1] 0.8055972
```

```
table(data$EmploymentEndReason,test_pred)
```

```
##    test_pred
##        0    1
##   0 1576   28
##   1  361   36
```

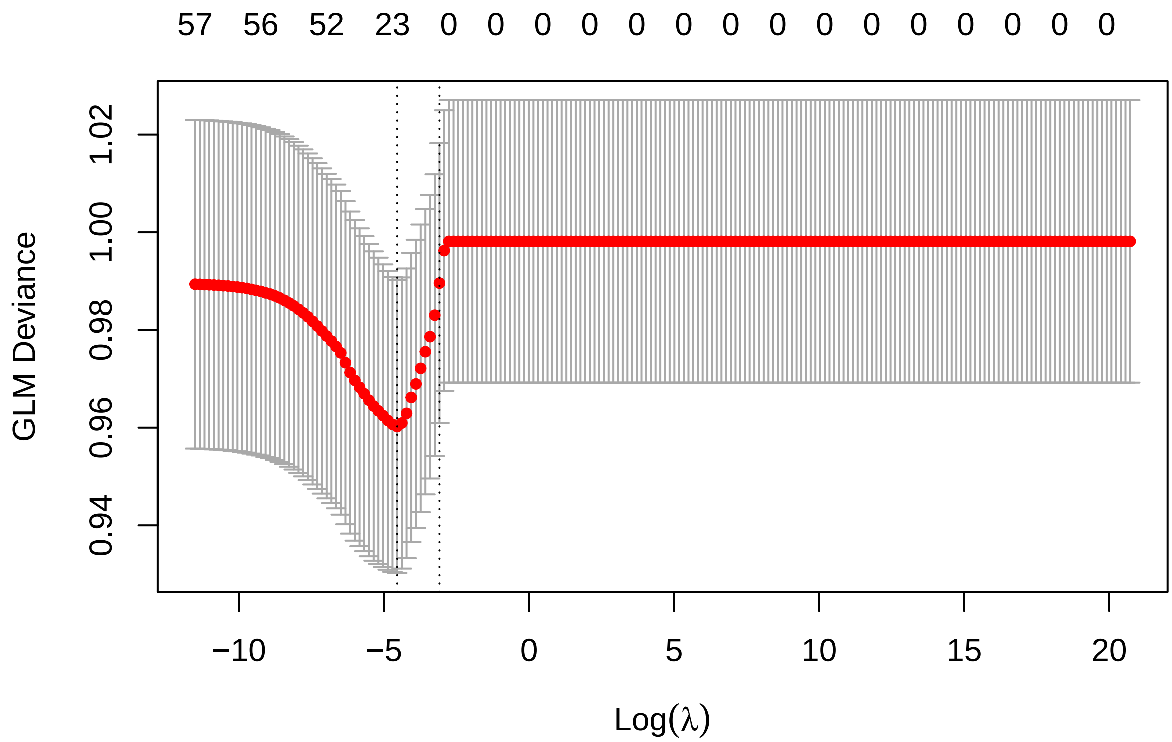
A simple logistic regression was run on the entire data set just to get an understanding of how a model would work on the data. The model had a training accuracy of ~80% which appears good on the surface. Looking at the confusion matrix we see the model was not able to separate the classes very well and ended up predicting the employee will stay the majority of the time. Because the classes are imbalanced, the model can maximize performance by just predicting 0.

Lasso Model

Perhaps there are too many predictors. A lasso model can eliminate variables that are not contributing by setting their coefficient to 0.

```
library(glmnet)

grid <- 10^seq(9, -5, length = 200)
lasso_model_cv <- cv.glmnet(X_train,Y_train,alpha = 1, lambda = grid,nfolds = 12, family = binomial)
plot(lasso_model_cv)
```




```
lasso_model_cv$lambda.min
```

```
## [1] 0.0105956
```

```
lasso_model <- glmnet(X_train,Y_train,alpha = 1, lambda = lasso_model_cv$lambda.min,nfolds = 12, family
lasso_prob <- predict(lasso_model,X_test,type = 'response')
lasso_pred <- ifelse(lasso_prob >.5, 1, 0)
mean(Y_test==lasso_pred)
```

```
## [1] 0.808
```

```
table(Y_test,lasso_pred)
```

```
##       lasso_pred
## Y_test   0   1
##      0 398   3
##      1  93   6
```

```
lasso_model$beta
```

```
## 59 x 1 sparse Matrix of class "dgCMatrix"
##                                              s0
## CommuteDistance                     0.023267644
## DegreeCompletedBachelor             .
## DegreeCompletedBelow College        .
## DegreeCompletedDoctor               0.323192930
## DegreeCompletedMaster               .
## DegreeFieldComputer Science         .
## DegreeFieldFinance                  .
```

```
## DegreeFieldLife Sciences          .
## DegreeFieldMarketing              .
## DegreeFieldNone                   .
## DegreeFieldOther                  .
## DegreeFieldTechnical Degree       .
## DepartmentHuman Resources         .
## DepartmentInformation Technology  .
## DepartmentMarketing               .
## DepartmentOther                   .
## DepartmentSales                   .
## YearsAtCompany                   -0.044748615
## GenderMale                       -0.067946669
## JobLevel2                         .
## JobLevel3                         0.126706718
## JobLevel4                         0.092297246
## JobLevel5                         .
## MaritalStatusMarried             -0.008741786
## MaritalStatusSingle               .
## MaritalStatusUnknown              0.635000976
## NumPreviousCompanies              0.064322788
## NumYearsWorked                    .
## OvertimeDays                      0.051473850
## OvertimeHours                     .
## Salary                            .
## TrainingsAttended                 .
## TravelFrequencyMonthly           -0.012885576
## TravelFrequencyNo Travel         -0.145794055
## TravelFrequencyNone               0.017647303
## TravelFrequencyWeekly             0.255984602
## Age                               0.020612323
## avg_review                        .
## sd_review                         .
## max_review                       -0.028564129
## min_review                        .
## Last_1_review                     .
## Q1.L                              .
## Q1.Q                              .
## Q1.C                              .
## Q1^4                              .
## Q2.L                              .
## Q2.Q                             -0.067506372
## Q2.C                              .
## Q2^4                              0.023672459
## Q3.L                              .
## Q3.Q                              .
## Q3.C                              .
## Q3^4                              .
## Q4.L                              .
## Q4.Q                             -0.168961996
## Q4.C                              .
## Q4^4                              .
## Has_no_review1                    .
```

The lasso model was able to set many of the coefficients to 0. We can assume what remains are the most impactful factors that predict which employees will leave. CommuteDistance and last review for example are still included in the model and the sign matches intuition. Surprisingly Salary was not included in the model. We can share these with the client as interesting insights. The model does not perform much better however. It still mainly predicts that the employee will stay on the test data set.

Logistic Model - Forward Selection

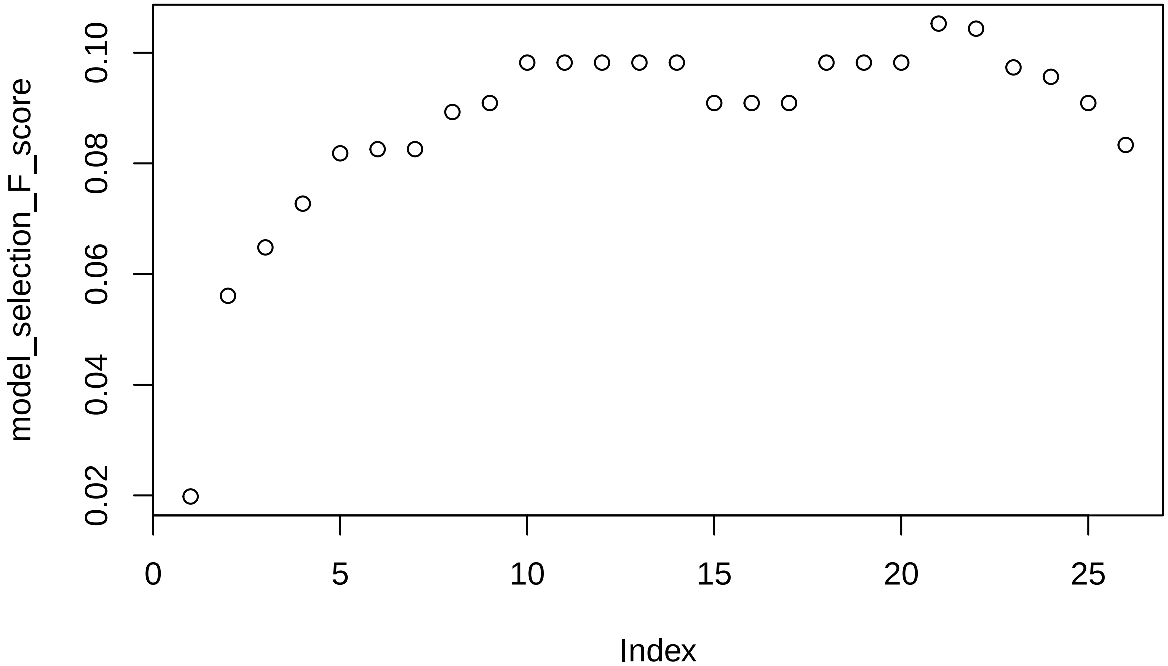
Logistic regression does not perform very well in a high dimensional setting. While the training set is ~1600, there are 26 variables, many of which have multiple levels. We will attempt to build a model using forward selection. Most importantly we will be using the F-score to evaluate the model. The F score in this instance is the harmonic mean of the recall and precision of the model and in an imbalanced class situation may be a better judge of model performance.

```r
## Forward Selection Bionomial
library(ROSE)

variables <- colnames(data)
variables <- variables[variables!='EmploymentEndReason' ]
best_F_log <- 0
best_model_logistic <- c()
model_selection_F_score <- rep(0,times = length(variables))
model_selection_var <- list()

for(i in c(1:length(variables))){
  if(i == 1){
    previous_model <- c()
  }else{
    previous_model <- model_selection_var
  }
  current_variables <- variables[!variables %in%previous_model]
  for(j in c(1:length(current_variables))){
    current_model <- unlist(append(previous_model,current_variables[j]))
    model_formula <- as.formula(paste('EmploymentEndReason ~ ',paste(current_model, collapse = '+')))
    logistic_model <- glm(model_formula,data = data_train,family = binomial)
    logistic_prob <- predict(logistic_model,data_test,type = 'response')
    logistic_pred <- ifelse(logistic_prob >.5,1,0)
    measures <- accuracy.meas(data_test$EmploymentEndReason,logistic_pred)
    if(is.na(measures$F) == FALSE){
      if(measures$F > model_selection_F_score [i]){
        model_selection_F_score[i] <- measures$F
        model_selection_var[i] <- current_variables[j]
      }
      if(measures$F  > best_F_log){
        best_F_log<- measures$F
        best_model_logistic <- current_model
      }
    }
  }
}

plot(model_selection_F_score)
```




```
best_model_logistic
```

```
##  [1] "OvertimeDays"         "NumPreviousCompanies" "max_review"
##  [4] "Q2"                   "TravelFrequency"      "Salary"
##  [7] "Age"                  "Q4"                   "DegreeField"
## [10] "Q1"                   "NumYearsWorked"       "OvertimeHours"
## [13] "TrainingsAttended"    "sd_review"            "Department"
## [16] "Last_1_review"        "Gender"               "Q3"
## [19] "Has_no_review"        "avg_review"           "DegreeCompleted"
```

```
model_formula <- as.formula(paste('EmploymentEndReason ~ ',paste(best_model_logistic, collapse = '+')))
logistic_model <- glm(model_formula,data = data_train,family = binomial)
logistic_prob <- predict(logistic_model,data_test,type = 'response')
logistic_pred <- ifelse(logistic_prob >.5,1,0)
table(data_test$EmploymentEndReason,logistic_pred)
```

```
##    logistic_pred
##       0   1
##   0 398   3
##   1  87  12
```

We see that 21 features were chosen as the best model. Any more and we cannot predict leaving with the same precision. The relevant features overlaps heavily with the coefficients of the lasso model.

KNN - Forward Selection

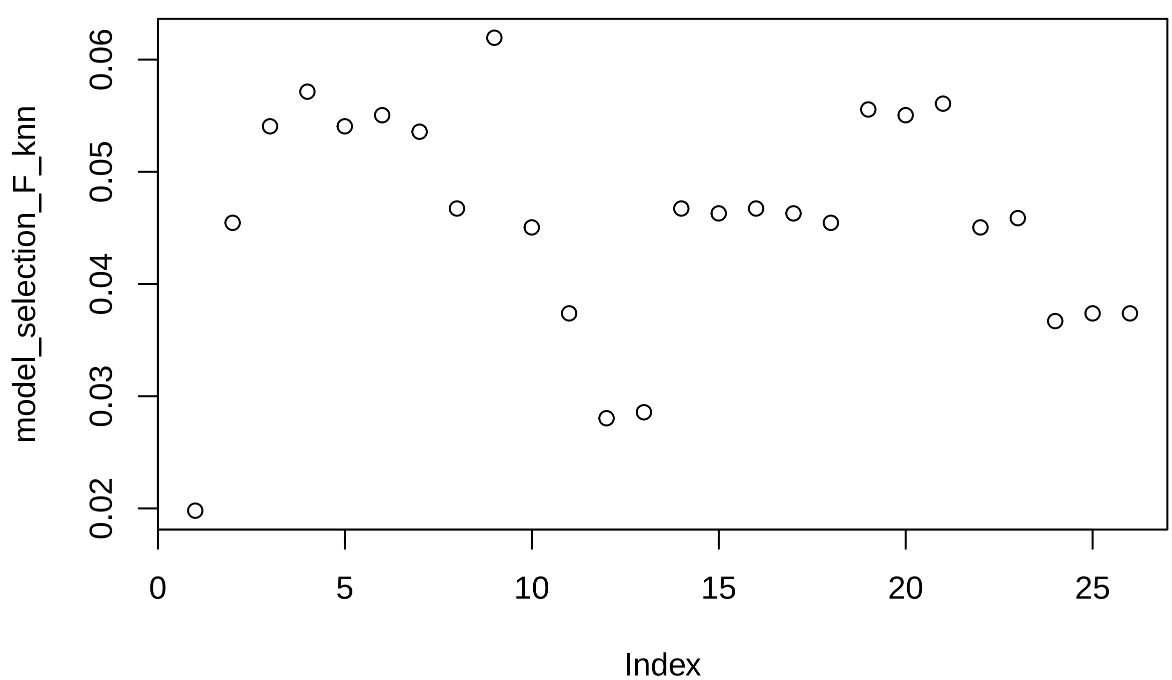
Just like logistic regression, KNN does not do well in high dimensional settings. We will also be attempting forward selection. The method is slightly different from above. Due to the time to train a KNN model, fitting

> 30 would take a very long time. Instead we fit the KNN model in order of significance determined by the logistic regression forward selection. This would introduce some bias, and a more in depth experiment would do a full forward selection.

```
## Forward Selection KNN
best_F_knn <- 0
best_model_KNN <- c()
model_selection_F_knn <- rep(0,times = length(model_selection_var))
time_start <- now()
for(i in 1:length(model_selection_var)){
  current_model <- model_selection_var[1:i]
  model_formula <- as.formula(paste('EmploymentEndReason ~ ',paste(current_model, collapse = '+')))
  KNN_model <- train(model_formula, data = data_train, method = 'knn',preProcess = c('center','scale'))
  KNN_prob <- predict(KNN_model,newdata = data_test,type = 'prob')
  KNN_pred <- ifelse(KNN_prob$'1' >.5,1,0)
  measures <- accuracy.meas(data_test$EmploymentEndReason,KNN_pred)
  model_selection_F_knn[i] <- measures$F
  if(measures$F  > best_F_knn){
    best_F_knn<- measures$F
    best_model_KNN <- current_model
  }
}
now() - time_start # ~3 minutes
```

```
## Time difference of 2.853824 mins
```

```
plot(model_selection_F_knn)
```



```r
best_model_KNN
```

```
## [[1]]
## [1] "OvertimeDays"
##
## [[2]]
## [1] "NumPreviousCompanies"
##
## [[3]]
## [1] "max_review"
##
## [[4]]
## [1] "Q2"
##
## [[5]]
## [1] "TravelFrequency"
##
## [[6]]
## [1] "Salary"
##
## [[7]]
## [1] "Age"
##
## [[8]]
## [1] "Q4"
##
## [[9]]
## [1] "DegreeField"
```

```r
model_formula <- as.formula(paste('EmploymentEndReason ~ ',paste(best_model_KNN, collapse = '+')))
KNN_model <- train(model_formula, data = data_train, method = 'knn',preProcess = c('center','scale'))
KNN_prob <- predict(KNN_model,newdata = data_test,type = 'prob')
KNN_pred <- ifelse(KNN_prob$'1' >.5,1,0)
table(data_test$EmploymentEndReason,KNN_pred)
```

```
##    KNN_pred
##       0   1
##   0 394   7
##   1  92   7
```

The forward selection algorithm has chosen the first 9 features for the KNN model.

Random Forest Model

The final model that will be tested is a random forest. The algorithm naturally samples from the total list of features so forward selection will not be needed.

```r
library(randomForest)
randforest_model <- randomForest(EmploymentEndReason~., data = data_train)
randforest_prob <- predict(randforest_model,newdata = data_test,type = 'prob')
randforest_pred <- ifelse(randforest_prob[,2] > .25,1,0)
best_F_forest <- accuracy.meas(data_test$EmploymentEndReason,randforest_pred)$F
table(data_test$EmploymentEndReason,randforest_pred)
```

```
##    randforest_pred
##       0   1
```

```
##    0 288 113
##    1  53  46
```

The random tree was able to separate the class to a much greater than the other models on the test data set.

Model Comparison

We will plot all model on an ROC Curve

```r
roc.curve(data_test$EmploymentEndReason,logistic_prob,main = 'Employee Turnover ROC Curve')
```

```
## Area under the curve (AUC): 0.643
```

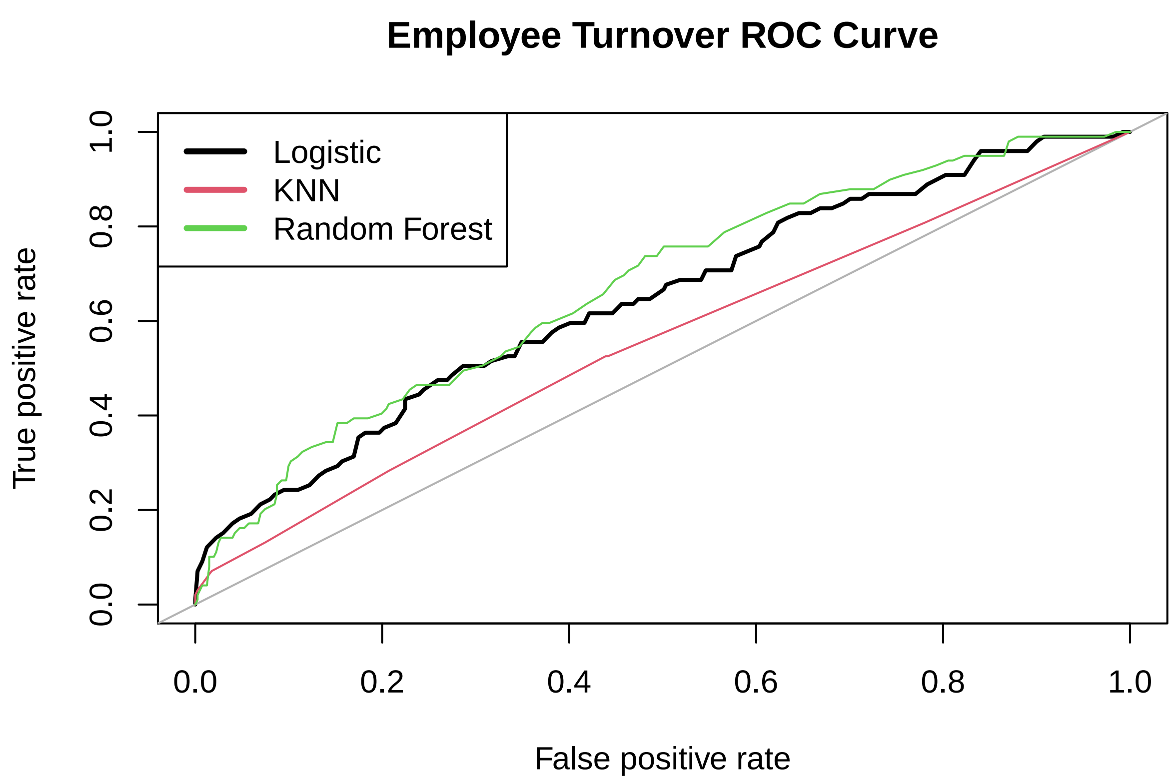
```r
roc.curve(data_test$EmploymentEndReason,KNN_prob$'1',add.roc = TRUE, col = 2)
```

```
## Area under the curve (AUC): 0.553
```

```r
roc.curve(data_test$EmploymentEndReason,randforest_prob[,2],add.roc = TRUE, col = 3)
```

```
## Area under the curve (AUC): 0.670
```

```r
legend('topleft',c('Logistic','KNN','Random Forest'),col = 1:3,lwd = 3)
```



It appears that random forest has the best performance by all metrics. The F-score is the greatest of the 3 models and it has the greatest AOC. AOC is a measure of accuracy, so the random forest it able to perform well overall and is able to separate the classes the best.

Sampling methods

One way to deal with imbalanced data is to create a new sample of data that is balanced in the classes. This can be done by oversampling (repeatedly add more samples of the smaller class) or undersampling (repeatedly removing samples of the larger class) until the final dataset is balanced.

```
data_train_over <- ovun.sample(EmploymentEndReason~., data = data_train, method = 'over')$data
data_train_under <- ovun.sample(EmploymentEndReason~., data = data_train, method = 'under')$data
```

With these new data sets we can train new models and compare the F-scores and ROC

Oversampling

```
randforest_model <- randomForest(EmploymentEndReason~., data = data_train_over)
randforest_prob <- predict(randforest_model,newdata = data_test,type = 'prob')
randforest_pred <- ifelse(randforest_prob[,2] > .25,1,0)

model_formula <- as.formula(paste('EmploymentEndReason ~ ',paste(best_model_KNN, collapse = '+')))
KNN_model <- train(model_formula, data = data_train_over, method = 'knn',preProcess = c('center','scale
KNN_prob <- predict(KNN_model,newdata = data_test,type = 'prob')
KNN_pred <- ifelse(KNN_prob$'1' >.5,1,0)

model_formula <- as.formula(paste('EmploymentEndReason ~ ',paste(best_model_logistic, collapse = '+')))
logistic_model <- glm(model_formula,data = data_train_over,family = binomial)
logistic_prob <- predict(logistic_model,data_test,type = 'response')
logistic_pred <- ifelse(logistic_prob >.5,1,0)

roc.curve(data_test$EmploymentEndReason,logistic_prob,main = 'Employee Turnover ROC Curve (Oversampling
```

```
## Area under the curve (AUC): 0.640
```

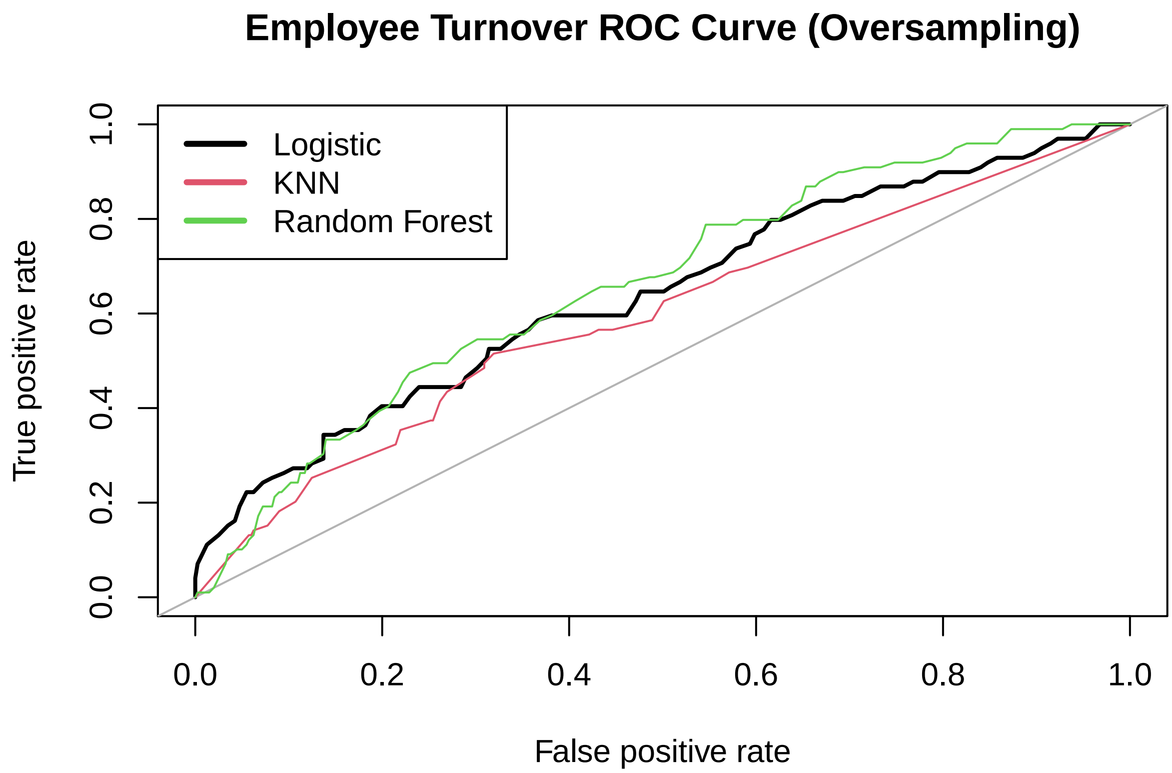
```
roc.curve(data_test$EmploymentEndReason,KNN_prob$'1',add.roc = TRUE, col = 2)
```

```
## Area under the curve (AUC): 0.593
```

```
roc.curve(data_test$EmploymentEndReason,randforest_prob[,2],add.roc = TRUE, col = 3)
```

```
## Area under the curve (AUC): 0.663
```

```
legend('topleft',c('Logistic','KNN','Random Forest'),col = 1:3,lwd = 3)
```




```
data.frame(logistic = accuracy.meas(data_test$EmploymentEndReason,logistic_pred)$F,
           knn = accuracy.meas(data_test$EmploymentEndReason,KNN_pred)$F,
           random_forest = accuracy.meas(data_test$EmploymentEndReason,randforest_pred)$F)
```

```
##    logistic       knn random_forest
## 1 0.1850534 0.1676647     0.1867816
```

Undersampling

```
randforest_model <- randomForest(EmploymentEndReason~., data = data_train_under)
randforest_prob <- predict(randforest_model,newdata = data_test,type = 'prob')
randforest_pred <- ifelse(randforest_prob[,2] > .25,1,0)

model_formula <- as.formula(paste('EmploymentEndReason ~ ',paste(best_model_KNN, collapse = '+')))
KNN_model <- train(model_formula, data = data_train_under, method = 'knn',preProcess = c('center','scal
KNN_prob <- predict(KNN_model,newdata = data_test,type = 'prob')
KNN_pred <- ifelse(KNN_prob$'1' >.5,1,0)

model_formula <- as.formula(paste('EmploymentEndReason ~ ',paste(best_model_logistic, collapse = '+')))
logistic_model <- glm(model_formula,data = data_train_under,family = binomial)
logistic_prob <- predict(logistic_model,data_test,type = 'response')
logistic_pred <- ifelse(logistic_prob >.5,1,0)

roc.curve(data_test$EmploymentEndReason,logistic_prob,main = 'Employee Turnover ROC Curve (Undersamplin
```

```
## Area under the curve (AUC): 0.674
```

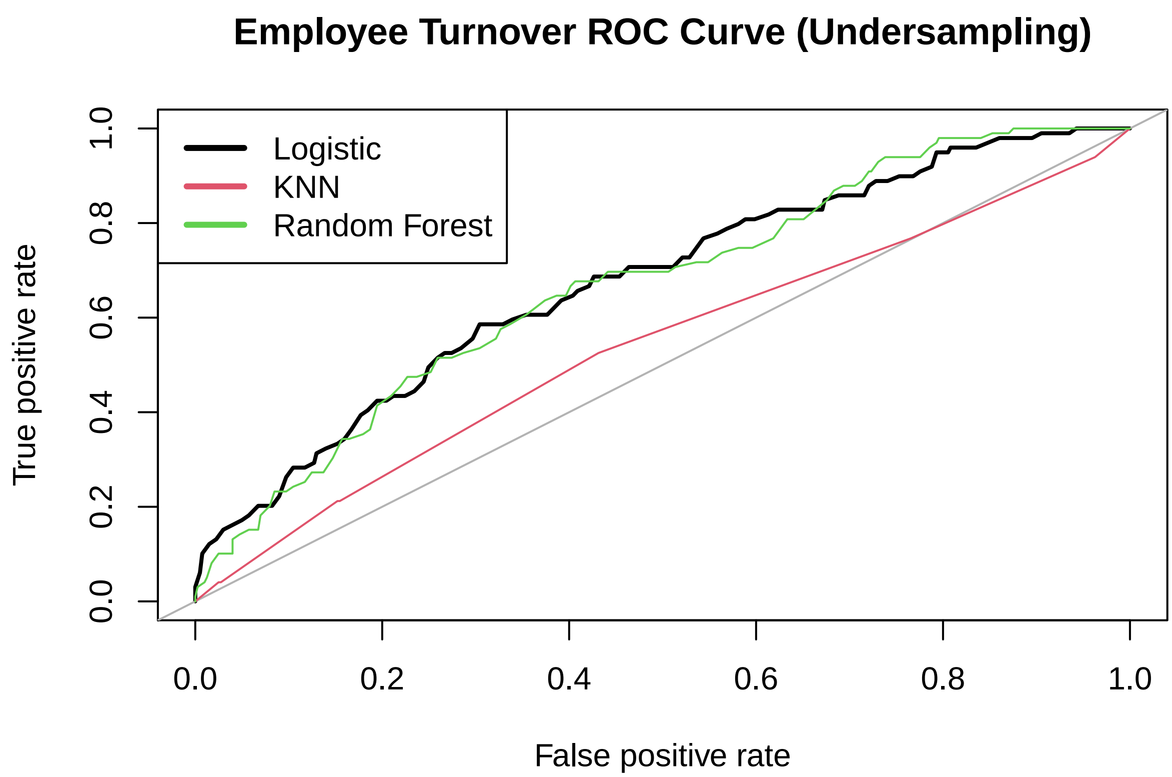
```r
roc.curve(data_test$EmploymentEndReason,KNN_prob$'1',add.roc = TRUE, col = 2)
```

```
## Area under the curve (AUC): 0.539
```

```r
roc.curve(data_test$EmploymentEndReason,randforest_prob[,2],add.roc = TRUE, col = 3)
```

```
## Area under the curve (AUC): 0.669
```

```r
legend('topleft',c('Logistic','KNN','Random Forest'),col = 1:3,lwd = 3)
```




```r
data.frame(logistic = accuracy.meas(data_test$EmploymentEndReason,logistic_pred)$F,
           knn = accuracy.meas(data_test$EmploymentEndReason,KNN_pred)$F,
           random_forest = accuracy.meas(data_test$EmploymentEndReason,randforest_pred)$F)
```

```
##   logistic       knn random_forest
## 1 0.192429 0.1604938          0.18
```

The sampling methods greatly improved the F-score for the logistic regression. The accuracy also improved as well and is fairly competitive with the random forest model in both the full and sampled data sets. The KNN model improved with the oversampling method. The tree performance remained mostly the same. Sampling can be a technique that can help improve the performance of machine learning in imbalanced datasets.