Predictive Classification of Employee Performance Reviews with Quadratic Discriminant Analysis

Word count: 1989

1. Introduction

This study looks to use a mixture of unsupervised learning, supervised learning, and analytic techniques in order to answer an inferential statistical research question pertaining to a dataset of employee information. In particular, the dataset consists of 1470 observations over 21 variables – both quantitative and qualitative – including monthly income, age, education, years of experience, and more. With a wealth of data available, statistical analysis and data science allows employers to gain new insights into the causal mechanisms that underly their organisation and their employee behaviour. It is these insights that are sought here.

2. Research Question

Among the variables contained within this dataset, there are a set of which that can be characterised as subjective descriptions given by both employee and employer. These include satisfaction scores given by employees and performance reviews given by employers – it is the latter of the two that is of primary concern here. Because these scores are subjective, this leaves open the possibility that they are not an accurate reflection of the employee performance which they are supposed to measure. This may be due to bias on the part of those performing the review, it may be a result of personal factors, or it may be some other unobservable factor. In any case, the subjective nature of the employee review leaves open the possibility that it is not a 'true' reflection of the employee's performance. As such, it would be beneficial to have derived a system by which one could classify an employee according to an *expected* performance review score, based on their characteristics. If such a system could be established with a good deal of efficacy and reliability, it could be measured against actual performance reviews, whereby any major and/or consistent departures may warrant further investigation. Thus, the research question at hand is: can an effective, reliable means of classifying employee performance based on their associated characteristics be developed?

3. Methods

First, I begin by employing *principal components analysis* (PCA) as an unsupervised learning technique (ULT)[1], in order to explore the data. PCA works by creating linear combinations – principal components (PCs) – of numeric variables, each of which explaining as much of the variance in the underlying data as possible. In doing so, a smaller number of features than the original dataset can be selected, while still capturing a large amount of the variance. Because the number of features/variables, or *dimensions*, is reduced, this is known as a method of *dimensionality reduction*.

Next, I use *quadratic discriminant analysis* (QDA) as a supervised learning technique. As the aim of this study is to develop a model that can accurately classify employees, a classification technique is most appropriate – QDA being one such technique. Its choice was motivated by two factors: first, our target variable (*PerformanceReview*) takes one of $k = 3$ values. This precludes us from using logistic regression, which is suited only to binary outcomes. Second, unlike *linear* discriminant analysis, QDA does not assume covariance matrices that are identical across classes. This leads to an altogether more flexible approach, that allows for cases in which variance of the inputs is not constant across outcome classes. Each of the above reasons make QDA well-suited to the research question at hand.

To measure the performance of the QDA model, this study uses the values for *accuracy, sensitivity,* and *specificity* generated from the model's resulting *confusion matrix*. A confusion matrix is a table that displays the number of observations with their actual and predicted values for each class. Accuracy is calculated as the number of correct predictions as a proportion of the overall predictions. Sensitivity describes a model's ability to correctly identify observations that belong to a certain class – the true positive rate. It is defined as:

$$Sensitivity = \frac{True\ positives}{True\ positives + False\ negatives}$$

Conversely, specificity describes the ability of a model to correctly identify observations that *don't* belong to a given class – the true negative rate. It is defined as:

$$Specificity = \frac{True\ negatives}{True\ negatives + False\ positives}$$

[1] ULTs refer to those methods which do not specifically define outcome variables of interest, and instead are used to explore the nature of a dataset with no prior assumptions about the relationships it contains. By contrast, supervised learning techniques rely on *pre-defined* outcome variables that are of interest.

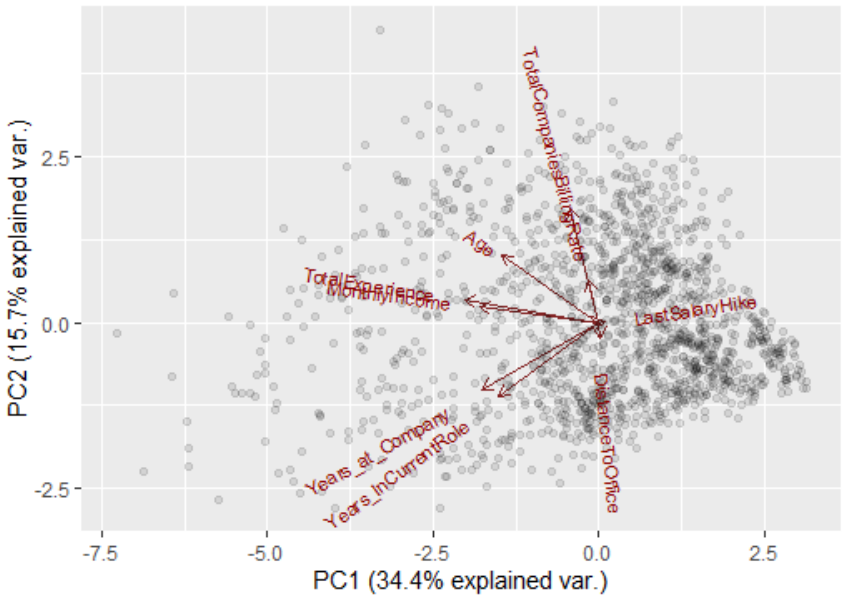
As a rule of thumb, the sum of specificity and sensitivity should be over 1.5, where a score of 2 is perfect, and a score of 1 is unusable[2]. Finally, construction of a QDA model is bolstered through the adoption of a *resampling method* as a means of model selection and evaluation.

Resampling refers to a process whereby samples are repeatedly drawn from the overall dataset and used to train/test the model, examining the results of each one. Used here is *k-fold cross-validation* (KFCV), which splits the data into k non-overlapping sets of training and validation data, using the training sets to estimate the model and the validation sets to test against.

4. Results

Analysis begins by running PCA on nine quantitative variables: number of companies worked for, years of experience, years at company, years in current role, billing rate, monthly income, most recent salary increase (%), distance from office to home, and age. With these variables, nine PCs are created, each of which accounting for a certain percentage of the total variance in the dataset. In this case, the first PC (*PC1*) explains 34.4% of the variance and is most strongly correlated with monthly income and years at company.

Figure 4.1: Biplot – Principal Components 1 & 2 (Descriptive)



[2] Power, M., Fell, G. and Wright, M. (2012). Principles for high-quality, high-value testing. *Evidence Based Medicine*, 18(1), p. 6

The second (*PC2*), explains 15.7% of the variance in the dataset, and is most strongly correlated with total companies, years in current role, and age. A biplot showing the scatter of variables and loadings of the variables for each PC is given by figure 4.1 above. The length and direction of the arrows indicates the *loading* of each constituent variable within the PC. So, for example, PC1 is a linear combination of the original variables that explains 34.4% of the variance in the dataset, and it is influenced strongly by: employee's years at company, their years in their current role, total experience, monthly income, and age. By contrast, their distance to the office is not strongly correlated with this PC. On the other hand, PC2 explains 15.7% of the variance and is most strongly correlated with the total number of companies worked at, and billing rate.

How many PCs should be included in further analysis is ultimately ambiguous, dependent on the trade-off between reducing the number of features and explaining the most variance. In this instance, the first five PCs are adopted, accounting cumulatively for 83.2% of variance (a scree plot showing cumulative variance across each PC is given in figure 6.1 in the appendix). Our analysis proceeds with these five components.

Next, a simple QDA model is built using a single sampling of the data for training and testing, whereby 85% of the data is randomly sampled for training, and the remaining 15% used to test. Prediction on the test data using this model yields mixed results, a sample confusion matrix of which is given by Figure 4.2 below. An intuitive way of reading a confusion matrix is that any entries sitting on the diagonal that runs from top left to bottom right are observations that have been correctly predicted.

Figure 4.2: Confusion Matrix (Inferential)

	Actual Class		
Predicted Class	Exceeds Expectations	Met expectations	Inconsistent
Exceeds Expectations	36	11	0
Met expectations	5	163	6
Inconsistent	0	0	0

With this sample, it is possible to calculate the model's *accuracy, sensitivity,* and *specificity* values. Calculating accuracy for the above model gives a value of 90%, initially suggesting strong predictive accuracy. For the outcome "Exceeds Expectations", the model gives a sum of 1.82 for sensitivity/specificity, suggesting a strong performance. For "Met Expectations", the same value is 1.70 – a weaker performance, but still relatively strong nonetheless. However, for the final target class – "Inconsistent" – the model gave a sum of sensitivity/specificity of 1, having essentially no predictive power, because the model did not assign any of the observations to this

class. Because the model did not assign any observations to this class, the number of true positives was zero, thus its sensitivity necessarily equals zero. Moreover, because the model predicted no positives for this class, it also predicted no false positives. Because, of this, the model's specificity is necessarily equal to 1. As a result, the value for mean sensitivity is brought down (0.60) and the value for mean specificity is raised (0.98). Considering only the mean values across the 2 other classes, the model has a slightly lower mean specificity (0.97) and a considerably higher mean sensitivity (0.91). This suggests that, in the absence of the "Inconsistent" outcome class, the model would have better overall performance. Regardless, in an attempt to improve this performance, KFCV – the resampling method outlined above – is used.

In this instance, $k = 10$ is selected, meaning that the model will be fit using 10 non-overlapping sets of training data, and is maximised according to the model's mean sensitivity. Here, the resulting model is essentially unchanged from previously. It has almost identical accuracy and means for sensitivity and specificity. This suggests that the resampling method above has not been able to resolve the issue of no observations being predicted for the "Inconsistent" class. This is because the dataset in question is heavily *imbalanced*, meaning the number of observations is heavily skewed across classes. Specifically, this class only represents 42 observations, compared to 1,185 for "Met Expectations". Classification models built on imbalanced datasets will naturally favour the outcome classes containing a greater number of observations in their predictions.

Overall, it is possible to construct a QDA model that performs relatively well in distinguishing between the two classes that represent a large portion of the datasets observations – "Exceeds Expectations" and "Met Expectations" – but that does not perform well when dealing with the remaining class.

5. Conclusion

Ultimately, the results of this study indicate that it is a promising line of questioning, but that the methods involved here were insufficient to build a model with the ideal level of predictive ability. There are two main drawbacks to the model that can be identified.

First, the use of QDA precludes the use of qualitative inputs/categorical independent variables. This is important because – as indicated by figures 6.2 to 6.5 in the appendix – there appears to be meaningful clusters of employees according to certain qualitative characteristics. As such, there are some categorical variables that may make useful explanatory variables, but are unsuited for the method chosen. For instance, both education and marital status seem to result in meaningful different clusters of employees when viewing their scatter according to our first two principal components. A more developed analysis would look to include these, and more, as features, in the hopes of increasing the number of explanatory variables. In particular, the adoption of a classifier that dealt well with discrete inputs, rather than continuous

– such as naïve Bayes classifier – would benefit this analysis.

Second, as described above, the model suffers from its application to an imbalanced dataset. Imbalanced classification is a well-known issue within inferential statistics, and there are a number of possible solutions. *Cost-sensitive learning* is a method whereby misclassifications on the minority class are 'penalised', giving this class greater consideration in the model's tuning. *Synthetic Minority Oversampling Technique* (SMOTE) uses data from the minority class to create new, synthetic examples on which to train the model. Either of the above methods, or any of the many others available, may be suitable in tackling the problem of imbalanced classification that has hampered this analysis.

This studies results suggest that – with the above problems being suitably addressed – it is possible to build a predictive classifier that is capable of solving the stated research problem.

6. Appendix

Figure 6.1: Scree Plot – Cumulative Variance Explained by Principal Components

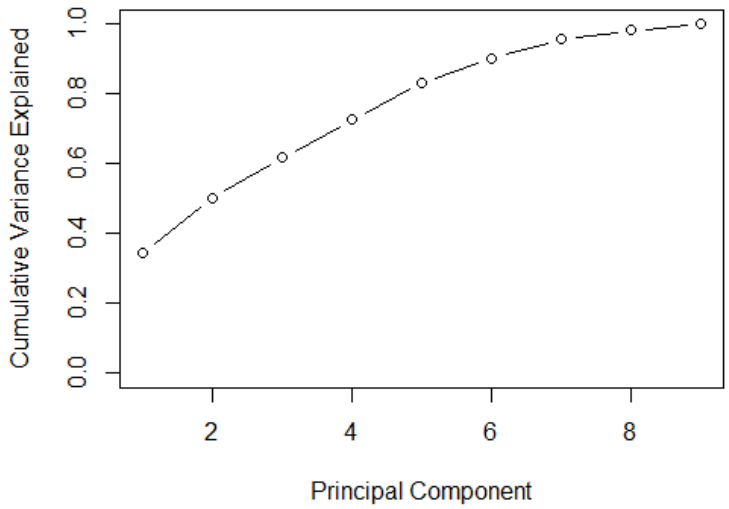


Figure 6.2 – Scatterplot of Employees by Principal Components 1&2, Clusters Highlighted According to Education

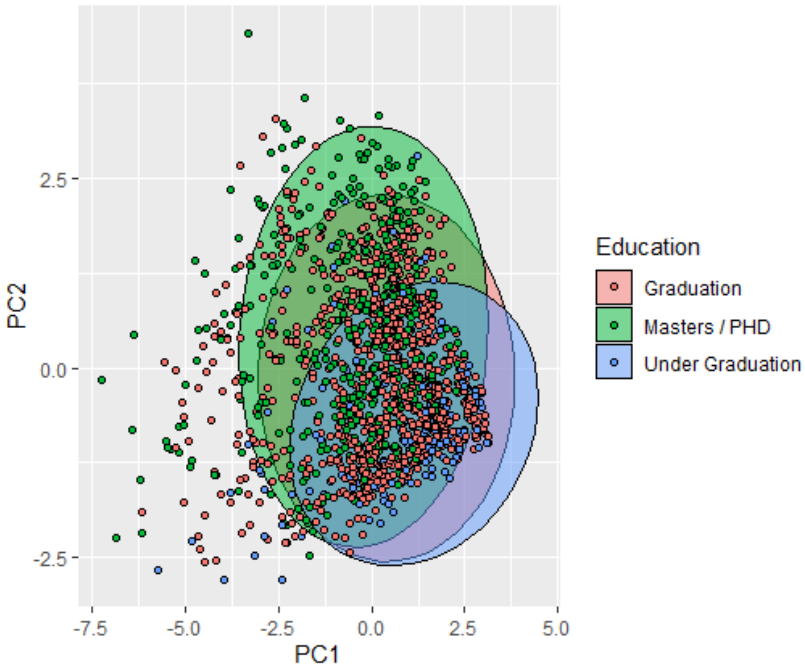


Figure 6.3 – Scatterplot of Employees by Principal Components 1&2, Clusters Highlighted According to Gender

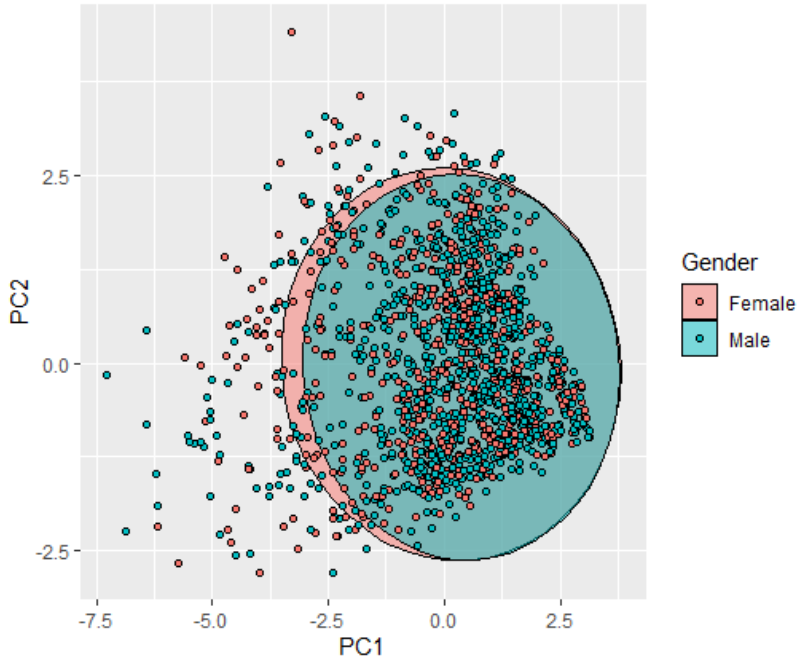
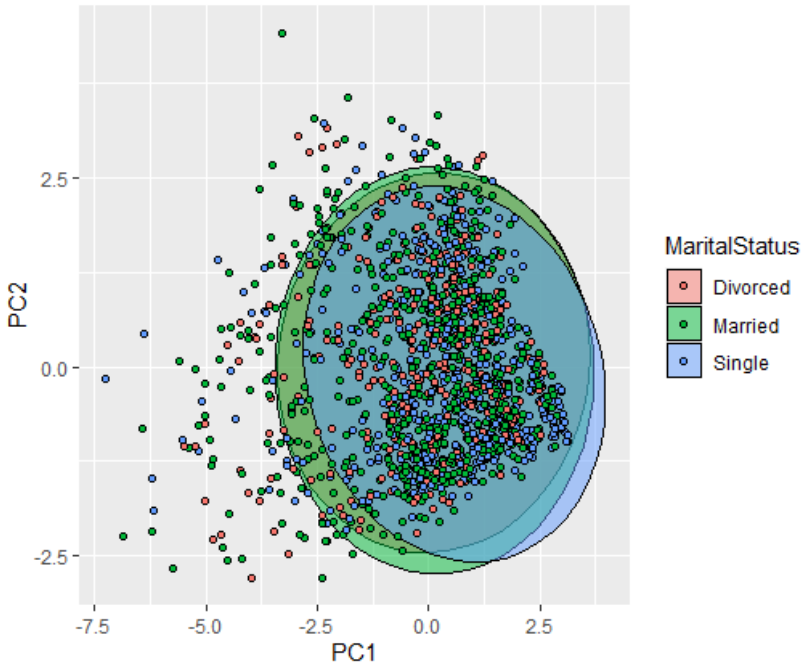


Figure 6.3 – Scatterplot of Employees by Principal Components 1&2, Clusters Highlighted According to Marital Status



Reproducible R Code

Below is an R code document allowing for the reproduction of the above results. Please note that, by convention, any packages that are required for installation are given as comments at the top of the script, so as not to inadvertently install anything unwanted. To install the packages, remove the hashtag from each before running.

```
# INSTALL PACKAGES AND LOAD LIBRARIES #

#install.packages("rlang")
#install_github('vqv/ggbiplot')
#install.packages('cluster')
#install.packages('factoextra')
#install.packages("caret")
#update.packages()

library(tidyverse)
library(readxl)
library(ggplot2)
library(devtools)
library(ggbiplot)
library(dplyr)
library(cluster)
library(factoextra)
library(MASS)
library(caret)

### EXPLORATORY DATA ANALYSIS AND UNSUPERVISED LEARNING

# IMPORT AND INSPECT DATA #

setwd("C:/Users/rocou/Desktop/Final Report/R") #set working directory
emp_data <- read.csv("employee_dataset.csv", sep=";") #read csv file to df

dim(emp_data) #21 variables, 1470 obs
names(emp_data) #variable names
head(emp_data) #initial 6 rows of data
sapply(emp_data, class) #variable data types
summary(emp_data) #descriptive summary of each variable



# REFORMAT DATA #

cols_num <- c('TotalCompanies','TotalExperience', 'DistanceToOffice',
              'BillingRate', 'MonthlyIncome', 'Years_at_Company',
              'Years_InCurrentRole', 'LastSalaryHike', 'Age') #select inte
ger columns

emp_data[cols_num] <- sapply(emp_data[cols_num], as.numeric) #turn integer
#columns into numeric
```

```
sapply(emp_data, class) #check data types



# PRINCIPAL COMPONENTS ANALYSIS AND VISUALISATIONS #

emp_data_pca <- subset(emp_data, select = c(TotalCompanies, TotalExperienc
e,
                                            DistanceToOffice, BillingRate,
                                            MonthlyIncome, Years_at_Compan
y,
                                            Years_InCurrentRole, LastSalar
yHike,
                                            Age))


apply(emp_data_pca, 2, mean, na.rm=TRUE) #this applies the mean function t
o each column
apply(emp_data_pca, 2, var, na.rm=TRUE)  #lots of difference in variance
# between variables, needs to be normalised


pr.out <- prcomp(na.omit(emp_data_pca), scale=TRUE)
names(pr.out)
pr.out$center   #means used for the scaling
pr.out$scale    #standard deviations used for the scaling (should be sqrt(
vars) from above)
pr.out$rotation #loading vectors
dim(pr.out$x)   #PC scores
summary(pr.out) #summary of PCs
names(pr.out)



ggbiplot(na.omit(pr.out), obs.scale = 1, var.scale = 1,  #plot PCs
         ellipse = FALSE, circle = FALSE, alpha = 0.1) + # reduce alpha fo
r viz
  scale_color_discrete(name = '') +
  theme(legend.direction = 'horizontal', legend.position = 'top')

pr.var <- pr.out$sdev^2 #define variance
pve <- pr.var / sum(pr.var) #define proportion of variance of each PC

plot(cumsum(pve),  #plot cumulative variance for all PCs
     xlab="Principal Component",
     ylab="Cumulative Variance Explained",
     ylim=c(0,1),
     type='b')

#cumulative sum graph indicates an (arbitrary) cut off of 5 PCs

emp_data2 <- cbind(emp_data, pr.out$x[,1:5]) #extract PCs

# Graph PC scores with highlighted selected employee characteristics
```

```
ggplot(emp_data2, aes(PC1, PC2, col=Education, fill=Education)) +
  stat_ellipse(geom="polygon", col="black", alpha=0.5) +
  geom_point(shape=21, col="black")

ggplot(emp_data2, aes(PC1, PC2, col=Gender, fill=Gender)) +
  stat_ellipse(geom="polygon", col="black", alpha=0.5) +
  geom_point(shape=21, col="black")

ggplot(emp_data2, aes(PC1, PC2, col=MaritalStatus, fill=MaritalStatus)) +
  stat_ellipse(geom="polygon", col="black", alpha=0.5) +
  geom_point(shape=21, col="black")

ggplot(emp_data2, aes(PC1, PC2, col=Overall_SatisfactionScore,
                      fill=Overall_SatisfactionScore)) +
  stat_ellipse(geom="polygon", col="black", alpha=0.5) +
  geom_point(shape=21, col="black")

#Convert our target/classes into factors

emp_data2 <- emp_data2 %>% mutate(PR_Factor = #create new numeric column
                     case_when(PerformanceReview=="Exceed Expectations" ~
0,
                               PerformanceReview == "Met Expectations" ~ 1
,
                               PerformanceReview == "Inconsistent" ~ 2))

emp_data2$PR_Factor <- as.factor(emp_data2$PR_Factor) #change numeric colu
mn
#to factor data type

sapply(emp_data2, class) #check data types


### QUADRATIC DISCRIMINANT ANALYSIS

# split dataset into testing and training components

emp_data2_split = sort(sample(nrow(emp_data2), nrow(emp_data2)*.85)) #

train <- emp_data2[emp_data2_split,]  #define training dataset
test <- emp_data2[-emp_data2_split,]  #define testing dataset

qda.model <- qda(PR_Factor~PC1+PC2+PC3+PC4+PC5, data=train) #train qda mod
el
qda.predict <- predict(qda.model, newdata=test, qda.model$prior) #predict
values
#on test dataset
conf_mat <- table(Predicted=qda.predict$class, Actual=test$PR_Factor) #con
fusion
#matrix
```

```
#Check accuracy, sensitivity, and specificity of model

accuracy = sum(conf_mat[1], conf_mat[5], conf_mat[9])/221 #calculate accur
acy as correct predictions/total pred.

#Calculate TP/TN/FP/FN for each class of sample conf matrix

c0_tp = conf_mat[1]
c0_tn = sum(conf_mat[5:6], conf_mat[8:9])
c0_fp = sum(conf_mat[4], conf_mat[7])
c0_fn = sum(conf_mat[2], conf_mat[3])

c1_tp = conf_mat[5]
c1_tn = sum(conf_mat[1], conf_mat[3],conf_mat[7],conf_mat[9])
c1_fp = sum(conf_mat[2], conf_mat[8])
c1_fn = sum(conf_mat[4], conf_mat[6])

c2_tp = conf_mat[9]
c2_tn = sum(conf_mat[1], conf_mat[2],conf_mat[4],conf_mat[5])
c2_fp = sum(conf_mat[3], conf_mat[6])
c2_fn = sum(conf_mat[7], conf_mat[8])

#Calculate sens./spec. for each class

c0_sens = c0_tp/(c0_tp+c0_fn)
c0_spec = c0_tn/(c0_tn+c0_fp)

c1_sens = c1_tp/(c1_tp+c1_fn)
c1_spec = c1_tn/(c1_tn+c1_fp)

c2_sens = c2_tp/(c2_tp+c2_fn)
c2_spec = c2_tn/(c2_tn+c2_fp)

mean_sens = (c0_sens+c1_sens+c2_sens)/3 #mean sensitivity across each clas
s
mean_spec = (c0_spec+c2_spec+c2_spec)/3 #mean specificity across each clas
s

mean_sens_01 = (c0_sens+c1_sens)/2 #sens. for classes 0 and 1
mean_spec_01 = (c0_spec+c2_spec)/2 #spec. for classes 0 and 1

### K-FOLD CROSS-VALIDATION

train.ctrl <- trainControl(method = "cv", number = 10, savePredictions=TRU
E,
                           summaryFunction = multiClassSummary)  #train co
ntrol
nb_fit <- train(PerformanceReview~PC1+PC2+PC3+PC4+PC5, data = emp_data2,
                method = "qda", trControl=train.ctrl, tuneLength = 0,
                metric='Sens') #fit model
nb_fit
```