Predicting the Absenteeism Hours at Workplace

Hong-Tan Lam & Ankit Tewari

Introduction

Linear Discriminant Analysis (LDA) is a well-established machine learning technique for predicting categories. Its main advantages, compared to other classification algorithms such as neural networks and random forests, are that the model is interpretable and that prediction is easy. Linear Discriminant Analysis takes a data set of cases (also known as observations) as input. For each case, you need to have a categorical variable to define the class and several predictor variables (which are numeric). We often visualize this input data as a matrix, such as shown below, with each case being a row and each variable a column. In this example, the categorical variable is called "class" and the predictive variables (which are numeric) are the other columns. Think of each case as a point in N-dimensional space, where N is the number of predictor variables. Every point is labeled by its category.

The LDA algorithm uses this data to divide the space of predictor variables into regions. The regions are labeled by categories and have linear boundaries, hence the "L" in LDA. The model predicts the category of a new unseen case according to which region it lies in. The model predicts that all cases within a region belong to the same category.

The linear boundaries are a consequence of assuming that the predictor variables for each category have the same multivariate Gaussian distribution. Although in practice this assumption may not be 100% true, if it is approximately valid then LDA can still perform well.

```
absenteeism <- read_delim("absenteeism_at_work.csv",
    ";", escape_double = FALSE, trim_ws = TRUE)
#absenteeism <- as_tibble(absenteeism)
absenteeism <- absenteeism[,-1]
absenteeism <- as.data.frame(absenteeism)
null=lm(absenteeism$`Absenteeism time in hours`~1, data=absenteeism)
full=lm(absenteeism$`Absenteeism time in hours`~., data = absenteeism)
result = step(null, scope=list(lower=null, upper=full), direction="both")
```

As we observe from the treat of our datasets through the stepwise variable selection, we can infer that the best AIC is observed for the model with the variables `Reason for absence` + `Disciplinary failure` + `Son` + `Height` + `Day of the week` + `Age` and `Absenteeism time in hours`. Therefore, we will take a subset of our dataset and begin modelling by considering these variables.

Methodology

Linear Discriminant Analysis is a supervised learning model that is similar to logistic regression in that the outcome variable is categorical and can therefore be used for classification. If you are unfamiliar with logistic regression, you can find a brief primer here.The difference between LDA and logistic regression is that LDA can be very useful when you are dealing with dealing with two response classes (although a multinomial logistic regression model can also be used in dealing with more than two response classes).

Whereas in logistic regression, where we model $P(Y = k|X = x)$ using the logistic function, in LDA we model the distribution of the predictors X separately in each of the response classes (i.e. given Y), and then use Bayes' Theorem to convert these into estimates for $P(Y = k|X = x)$. When these distributions are assumed to be normally distributed, it turns out that the model is very similar in form to logistic regression. If this is the case, you may be wondering why not use logistic regression? LDA provides three advantages over logistic regression.

When the classes are well-separated, the parameter estimates for the logistic regression model are fairly unstable. LDA does not suffer from this problem. If n is small, and the distribution of the predictors X is approximately normal in each of the classes, the linear disriminant model is again more stable than the logistic regression model. LDA is popular when we have more than two response classes.

Using Bayes' Theorem for Classification

Let us suppose that we would like to classify an observation into on of K classes. Let π_k denote the prior probability that a randomly chosen observation comes from the kth class. this is the probability that a randomly chosen observation is associated with the kth category of the response variable Y. Then, let $f_k(X) = Pr(X = x|Y = k)$ denote the density function of X for an observation that comes from the kth class. That is, $f_k(x)$ is relatively large if there is a high probability that an observation in the kth class has $X \sim x$ and fk(x) is small if it is very unlikely that an observation in the kth class has $X \sim x$. Then, Bayes' Theorem states that

$$P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

Our goal is to estimate π_k and $f_k(X)$. The former is relatively easy if we have a random saple of $Y's$ from the population. We simply compute the fraction of the training observations the belong to the kth class. However, estimating $f_k(X)$ tends to be more difficult, unless we assume some simple forms for these densities. We refer to $P_k(x) = P(Y = k|X = x)$ as the posterior probability that an observation $X = x$ belongs to the kth class.

Linear Discriminant Analysis for p=1

Let us assume that $p = 1$, which means that we have only one predictor. Our goal is to obtain an estimate for $f_k(x)$ that we can plug into the above equation in order to estimate $p_k(x)$. We will then classify an observation to the class for which $p_k(x)$ is the greatest. To estimate $f_k(x)$, we will need to make a few assumptions about its form.

If we assume that $f_k(x)$ is normally distributed, in a situation where $p = 1$, the normal density takes the form-

$$f_k(x) = \frac{1}{(2\pi\sigma_k^2)^{1/2}} exp\{\frac{-1}{2}\sigma_k^2(x - \mu_k)^2\}$$

where μ_k and σ_k^2 are the mean and variance parameters for the kth class, respectively. For now, let us assume that there is a shared variance term across all K classes $\sigma_1^2, ..., \sigma_K^2$. For simplicty, we can denote it with σ^2. We then get

$$p_k(x) = \frac{\pi_k.\{\frac{-1}{2}\sigma^2(x - \mu_k)^2\}}{\sum_{i=1}^{l} \pi_l\{\frac{-1}{2}\sigma^2(x - \mu_l)^2\}}$$

If we take the logarithm of the above equation and rearrange the terms, we can show that this is equivalent to assigning the observation to the class for which

$$\delta_k = x.\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + log(\pi_k)$$

is largest.

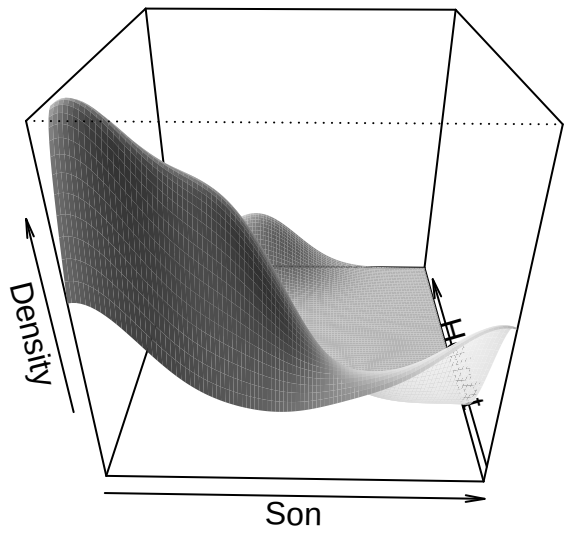
Now, we will read the data and perform the downsampling in order to allow the dataset to contain relevant number of observations by means of simple random sampling. The next few lines also convert the target variable into a factor with levels (0-29), (30-59) and (60+)

```r
absenteeism <- absenteeism[order(absenteeism$`Absenteeism time in hours`),]

downsampling <- downSample(as.matrix(absenteeism[,1:19]), as.factor(absenteeism[,20]))
downsampling <- downsampling[,c(1,3,11,13,18,20)]

hours.cat <- function(x, lower = 0, upper, by = 10,
                      sep = "-", above.char = "+") {

 labs <- c(paste(seq(lower, upper - by, by = upper+1),
                 seq(lower + by - 1, upper - 1, by = by),
                 sep = sep),
           paste(upper, above.char, sep = ""))

 cut(floor(x), breaks = c(seq(lower, upper, by = by), Inf),
     right = FALSE, labels = labs)
}

names(downsampling)[6] <- c("Absenteeism time in hours")
downsampling$`Absenteeism time in hours` <-
  as.numeric(as.character(downsampling$`Absenteeism time in hours`))
downsampling$`Absenteeism time in hours`<-
  hours.cat(downsampling$`Absenteeism time in hours`, upper = 80, by=30)
```

```r
set.seed(123)

#downsampling <- downSample(absenteeism, as.factor(absenteeism[,6]))
sampled <- sample(c(TRUE, FALSE), nrow(downsampling), replace = T, prob = c(0.75,0.25))
absent_train <- as.data.frame(downsampling[sampled, ])
absent_test <-  as.data.frame(downsampling[!sampled,])
#upsampling <- upSample(absent_train[,-6], as.factor( absent_train[,6]))
```

We now perform the Mulivariate normality inference which is a prerequisite for the LDA-

```r
mvn_test <- mvn(data = absent_train[,4:5], mvnTest = "mardia", multivariatePlot = "persp")
```

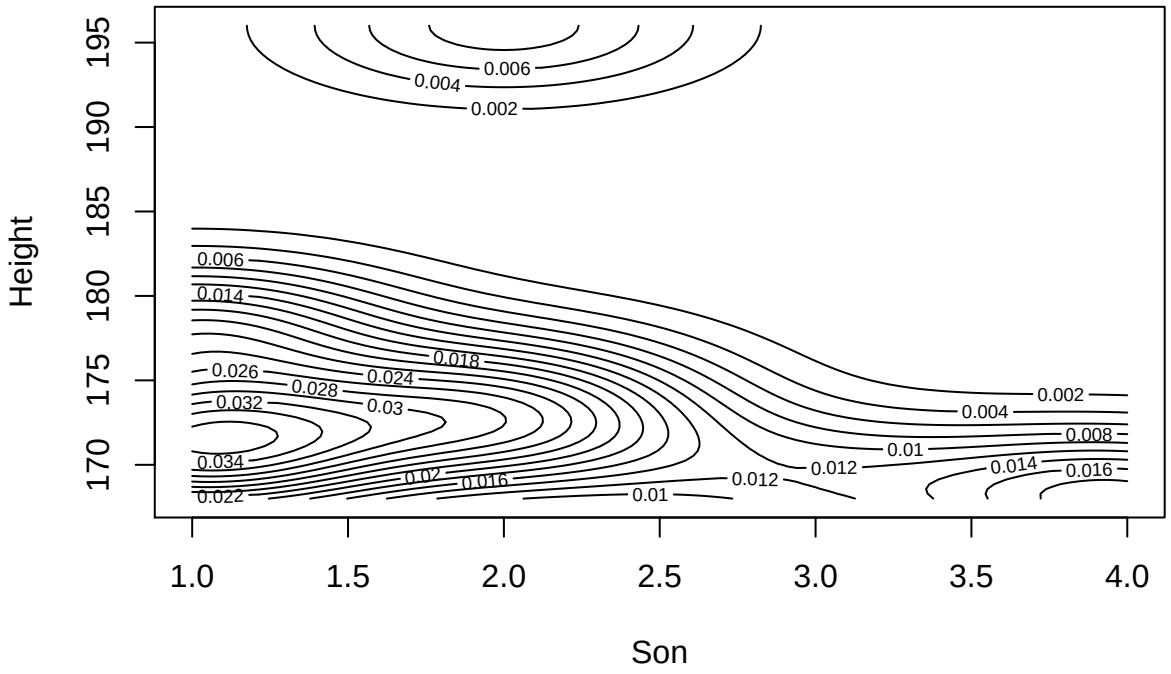



```r
print(mvn_test)
```

```
## $multivariateNormality
```

```
##              Test        Statistic              p value Result
## 1 Mardia Skewness 22.5653368891208 0.000154629404068713     NO
## 2 Mardia Kurtosis 1.48689411768738    0.137042778110413    YES
## 3             MVN             <NA>                  <NA>     NO
##
## $univariateNormality
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk    Son       0.7847    0.0045    NO
## 2 Shapiro-Wilk  Height      0.7051    0.0006    NO
##
## $Descriptives
##         n       Mean  Std.Dev Median Min Max 25th 75th      Skew
## Son    13   1.923077 1.115164      2   1   4    1    2 0.8014933
## Height 13 173.846154 7.436914    172 168 196  169  175 1.9341862
##          Kurtosis
## Son    -0.8653851
## Height  3.1616671
```

```
mvn_test <- mvn(data = absent_train[,4:5], mvnTest = "mardia", multivariatePlot = "contour")
```




```
print(mvn_test)
```

```
## $multivariateNormality
##              Test        Statistic              p value Result
## 1 Mardia Skewness 22.5653368891208 0.000154629404068713     NO
## 2 Mardia Kurtosis 1.48689411768738    0.137042778110413    YES
## 3             MVN             <NA>                  <NA>     NO
##
## $univariateNormality
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk    Son       0.7847    0.0045    NO
## 2 Shapiro-Wilk  Height      0.7051    0.0006    NO
##
## $Descriptives
```

```
##          n       Mean  Std.Dev Median Min Max 25th 75th      Skew
## Son     13   1.923077 1.115164      2   1   4    1    2 0.8014933
## Height 13 173.846154 7.436914    172 168 196  169  175 1.9341862
##          Kurtosis
## Son     -0.8653851
## Height  3.1616671
```

Since the variable 3 seems to be constant across all observations, we remove it.

```
lda.m1 <- lda(absent_train$`Absenteeism time in hours`~. , data = absent_train)
```

```
#absent_test <- absent_test[, c(-3,-6)]
test.predicted.lda <- predict(lda.m1, newdata = absent_test[,-6])
```

Results

We now discuss the results obtained using the application of LDA model on our dataset in the following subsections as follows-

Confusion Matrix

```
lda.cm <- table(absent_test$`Absenteeism time in hours`, test.predicted.lda$class)
list(LDA_model = lda.cm %>% prop.table() %>% round(3))
```

```
## $LDA_model
##
##         0-29  0-59   80+
##   0-29 0.333 0.000 0.333
##   0-59 0.000 0.000 0.167
##   80+  0.000 0.167 0.000
```

```
#     QDA_model = qda.cm %>% prop.table() %>% round(3))
diag(prop.table(lda.cm))
```

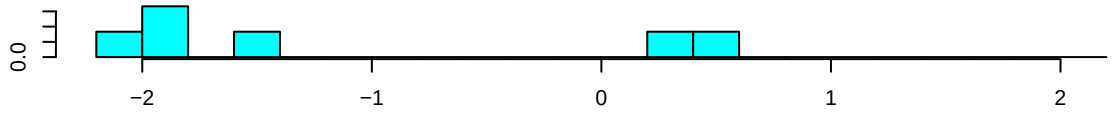
```
##      0-29      0-59       80+
## 0.3333333 0.0000000 0.0000000
```

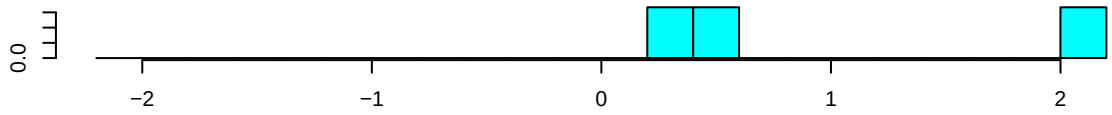
```
sum(diag(prop.table(lda.cm)))
```

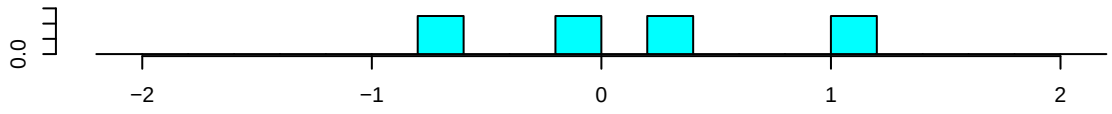
```
## [1] 0.3333333
```

The confusion matrix above illustrates quite clearly that the we have successfully classified the data in the two of three classes. Although, the model fails in terms if discriminating the second class from the first and third and can be seen as a drawback. This point is discussed in the discussion part.

```
plot(lda.m1, dimen=1)
```



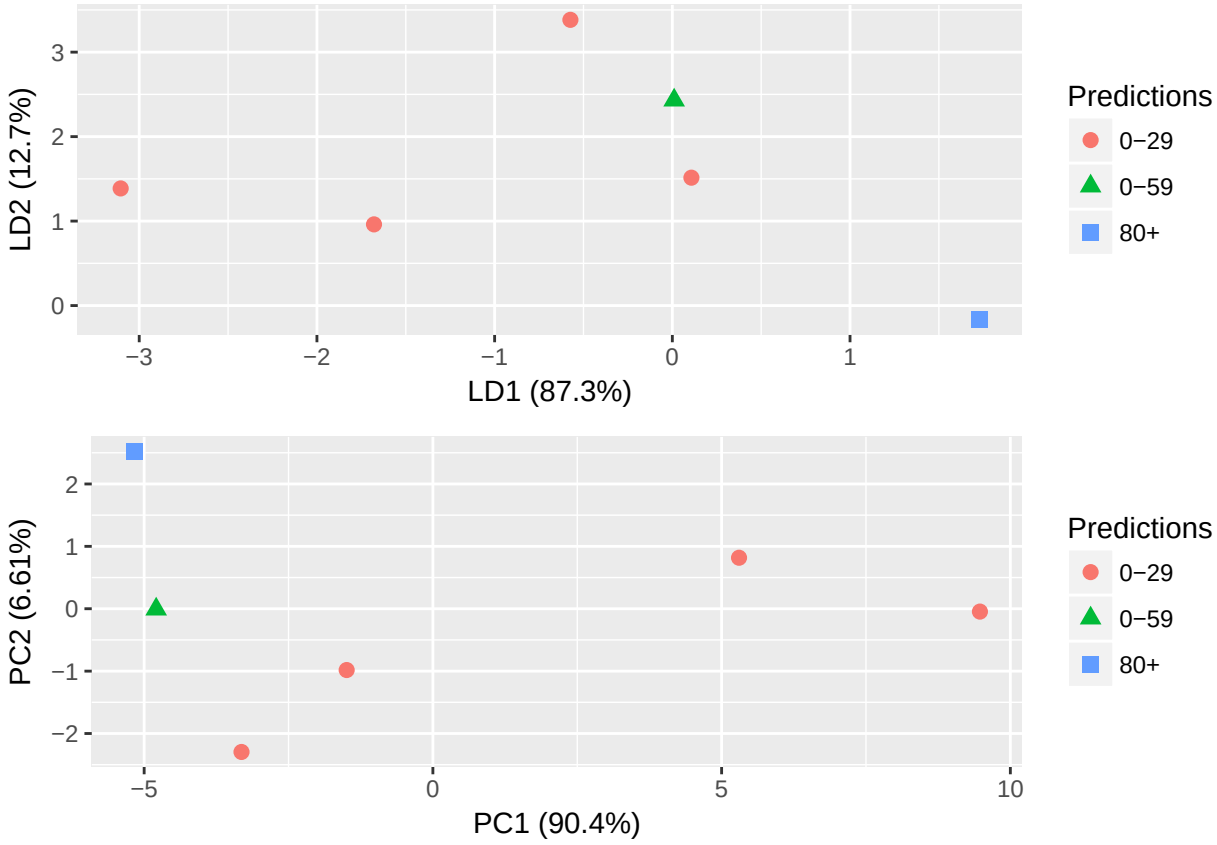




We can observe from these histograms that the model has learnt significant discriminatice capacity despite of the fact that the training dataset provided was small in size.

```
require(MASS)
require(ggplot2)
require(scales)
require(gridExtra)


#histogram(df.pred$posterior[,1])
#histogram(df.pred$posterior[,2])
#histogram(df.pred$posterior[,3])

#names(absent_test)[6] <- c("Predictions")

pca <- prcomp(absent_test[,c(-3,-6)],
              center = TRUE,
              scale. = F)

prop.pca = pca$sdev^2/sum(pca$sdev^2)

prop.lda = lda.m1$svd^2/sum(lda.m1$svd^2)
dataset <- data.frame(Predictions = absent_test[,6], pca = pca$x,
                      lda = test.predicted.lda$x)



p1 <- ggplot(dataset) + geom_point(aes(lda.LD1, lda.LD2, colour = Predictions,
```

```
                                    shape = Predictions), size = 2.5) +
  labs(x = paste("LD1 (", percent(prop.lda[1]), ")", sep=""),
       y = paste("LD2 (", percent(prop.lda[2]), ")", sep=""))

p2 <- ggplot(dataset) + geom_point(aes(pca.PC1, pca.PC2, colour = Predictions,
                                    shape = Predictions), size = 2.5) +
  labs(x = paste("PC1 (", percent(prop.pca[1]), ")", sep=""),
       y = paste("PC2 (", percent(prop.pca[2]), ")", sep=""))
grid.arrange(p1, p2)
```



Discussions

As we can clearly observe that the model fitted using the Linear Discriminant Analysis method had a clearly good capability to classify the observations in the class 1 (i.e. absenteesim time of 0-29 hours) and to some extent the observations of class 3 (60+) also. However, some points regarding the model are worth noting such as-

a) the model clearly lagged in terms of predicting the observations of the class (30-59). One of the reasons attributed to this is the idea of downsampling that was adopted to preprocess the dataset had resulted in one of the columns getting filled with all zeros and hence becoming constant throughtout which created problem in fitting the LDA model. Eventually, this column, named disciplinary failure was dropped.

b) The idea of Upsampling was proposed initially but was brought into consideration while fitting the model. However, the model was suffering throught he problem of singularity on fitting using the upsampling approach. In that case, some steps will be taken in future to ensure implementation of

this approach for e.g. the application of upsampling procedure prior to application of variable selection might serve our purpose which was done in reverse order in the present case.

c) Finally, we were unable to fit a Quadratic Discriminant Analysis model as one of the classes of the dataset was having still lesser observations than required by the QDA for fitting the model. We will explore newer ways of fitting the QDA model by improvised pre-processing of the dataset.

Conclusions

The problem of identifying and modelling the employee absence rate at workplace requires further brainstorming with many other machine learning methods so that we can make critical comparisons. Random forests and Support vector regression is proposed as one of the preferable approaches for dealing with the problem without transforming the output variable to a factor and hence the regression problem to its classification counterpart.

Such methods and their application on these datasets present a great hope in the nearby future where we can see more employee friendly emerging with the help of machine learning methods.

References

- Linear Disriminiant Analyis- An Introduction, R Bloggers
- Computing and Visualizing Linear Discriminant analysis model in R
- S Korkmaz, D Goksuluk, and G Zararsiz. Mvn: An r package for assessing multivariate normality. The R Journal, 6(2):151–162, 2014.
- [Tom Burdenski. Evaluating univariate, bivariate, and multivariate normality using graphical and statistical procedures. Multiple Linear Regression Viewpoints, 26(2):15–28, 2000]
- [James P Stevens. Applied multivariate statistics for the social sciences. Routledge, 2012]
- Addressing Multicollinearity in R: An approach based on VIF and similar methods