Develop a Linear Discriminant Analysis model to classify the LeaveOrNot event from the other variables.

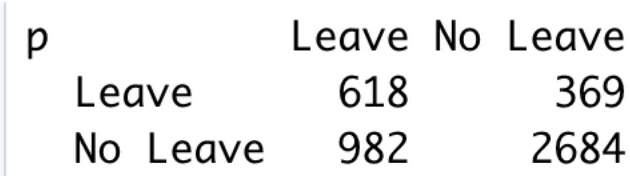
```
#Set working directory
setwd("~/Desktop/Stats Fall 2021")

#Read in dataset
dataset <- read.csv(file="Employee.csv", header=TRUE, sep=",")

#Check Sample Size and Number of Variables
dim(dataset)
# 4653 sample size and 9 variables

#Check Missing Data
sum(is.na(dataset))
# No missing values.

head(dataset)

tail(dataset)

View(dataset)

str(dataset, list.len=ncol(dataset))
str(dataset)
# String variables: Education, City, Gender, EverBenched

# Convert string variables to numeric for regression modeling

library(plyr)

#Gender

table(dataset$Gender)
dataset$Gender_num <- revalue(dataset$Gender, c("Male"="1", "Female"="0"))
dataset$Gender_num <- as.numeric(dataset$Gender_num)

#EverBenched

table(dataset$EverBenched)
```

```
dataset$EverBenched_num <- revalue(dataset$EverBenched, c("Yes"="1", "No"="0"))
dataset$EverBenched_num <- as.numeric(dataset$EverBenched_num)

#Education

table(dataset$Education)
dataset$Education_num <- revalue(dataset$Education, c("Bachelors"="0",
"Masters"="1","PHD"="2" ))
dataset$Education_num <- as.numeric(dataset$Education_num)

#City

table(dataset$City)
dataset$City_num <- revalue(dataset$City, c("Bangalore"="0", "New Delhi"="1","Pune"="2" ))
dataset$City_num <- as.numeric(dataset$City_num)

# Age

table(dataset$Age)

dataset$Age[(dataset$Age >=22) & (dataset$Age<=26)] <- 0
dataset$Age[(dataset$Age >=27) & (dataset$Age<=31)] <- 1
dataset$Age[(dataset$Age >=32) & (dataset$Age<=36)] <- 2
dataset$Age[(dataset$Age >=37) & (dataset$Age<=41)] <- 3

# Year

dataset$JoiningYear[dataset$JoiningYear == 2012] <- 0
dataset$JoiningYear[dataset$JoiningYear == 2013] <- 1
dataset$JoiningYear[dataset$JoiningYear == 2014] <- 2
dataset$JoiningYear[dataset$JoiningYear == 2015] <- 3
dataset$JoiningYear[dataset$JoiningYear == 2016] <- 4
dataset$JoiningYear[dataset$JoiningYear == 2017] <- 5
dataset$JoiningYear[dataset$JoiningYear == 2018] <- 6

# LeaveOrNot

library(dplyr)
dataset$LeaveOrNot = as.factor(dataset$LeaveOrNot)
dataset$LeaveOrNot <- recode_factor(dataset$LeaveOrNot, "1"="Leave", "0"="No Leave")

# Subset num categories into new dataframe

str(dataset)
```

```
colnames(dataset)

num_data <- dataset[,c(2, 4,5,8:13)]

# Double check new dataset
str(num_data)
# 4653 sample size and 9 numeric variables
```

'Q1) Develop a Linear Discriminant Analysis model to classify the LeaveOrNot event from the other variables.'
'a) What is the performance of the classifier using cross-validation?'

```
library(MASS)

head(num_data)

#With Cross Validation
# The dependent variable must be categorical
LeaveOrNotLDA <- lda(LeaveOrNot ~ ., data=num_data, CV=TRUE)
LeaveOrNotLDA

#To Plot the Data, you cannot use CV
LeaveOrNotLDA <- lda(LeaveOrNot ~ ., data=num_data)
LeaveOrNotLDA

plot(LeaveOrNotLDA, xlab = "LD1", ylab = "LD2")

# Try to predict the class from the original data
colnames(num_data)
p <- predict(LeaveOrNotLDA, newdata=num_data)$class
p

# Compare the results of the prediction (Confusion Matrix)
table1<-table(p, num_data$LeaveOrNot)
table1
```



```
p            Leave No Leave
  Leave        618      369
  No Leave     982     2684
```

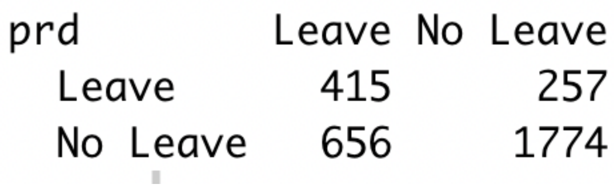
```
#Using Trace
```

```
sum(diag(table1)/sum(table1))

# Accuracy
accuracy <- ((618+2684)/(982+369+618+2684))
accuracy
#0.71

mean(p== num_data$LeaveOrNot)
#0.71
```

'ANS 1a: The performance of the classifier is 71% using cross-validation. The model suggests that after two years, 70.7 percent of employees leave the company.'

'b) What is the performance of the classifier using training and testing?'

```
#Creating Training and Testing Samples
require(caTools)  # loading caTools library
library(caTools)
set.seed(123)   #  set seed to ensure you always have same random numbers generated
sample = sample.split(num_data,SplitRatio = 0.70)
train =subset(num_data,sample ==TRUE) # creates a training dataset named train with rows which are marked as TRUE
test=subset(num_data, sample==FALSE)

# The dependent variable must be categorical (Assuming No Cross-Validation)
LeaveOrNotLDA1 = lda(LeaveOrNot ~ ., data=train)
LeaveOrNotLDA1

plot(LeaveOrNotLDA1)

# Try to predict the class from the original data
prd<-predict(LeaveOrNotLDA1, train)

# Compare the results of the prediction
table2 = table(p, train$LeaveOrNotLDA1)
table2
```



```
prd         Leave No Leave
   Leave      415      257
   No Leave   656     1774
```

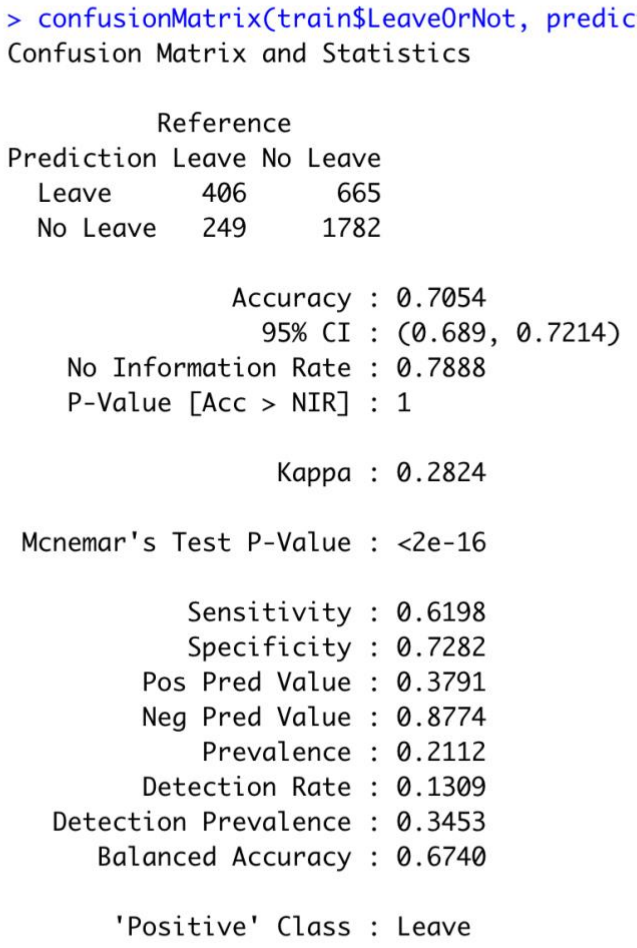
#Using Trace
sum(diag(table2)/sum(table2))

'ANS 1b: Performance of classifier using training and testing is 71%. The training and testing data select random samples from the dataset using a 70/30 ratio. There are 415 true positives, 257 false positive (type 2 error) , 656 false negatives (type 1 error) and 1774 true negatives detected in this sample. '

'C. Would certain misclassification errors be worse than others? If so, how would you suggest measuring this? '

library(caret)
modelFit<- train(LeaveOrNot ~ ., method='lda',preProcess=c('scale', 'center'), data=train)

#Confusion Matrix
confusionMatrix(train$LeaveOrNot, predict(modelFit, train))

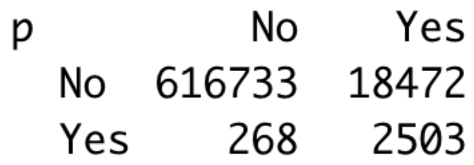


```
> confusionMatrix(train$LeaveOrNot, predict(modelFit, train))
Confusion Matrix and Statistics

          Reference
Prediction Leave No Leave
  Leave      406      665
  No Leave   249     1782

               Accuracy : 0.7054
                 95% CI : (0.689, 0.7214)
    No Information Rate : 0.7888
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2824

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6198
            Specificity : 0.7282
         Pos Pred Value : 0.3791
         Neg Pred Value : 0.8774
             Prevalence : 0.2112
         Detection Rate : 0.1309
   Detection Prevalence : 0.3453
      Balanced Accuracy : 0.6740

       'Positive' Class : Leave
```

“ANS 1c: Yes, Type 1 and Type 2 errors are some important misclassification errors. In this case it is important to reduce both false positives (larger predicted value of people leaving) and false negatives (over estimate of people not leaving). These errors are more important than other metrics like accuracy and precision. The sensitivity helps check for misclassification error. The model correctly classfied an employee who was about to leave the company with a success of ~62%.

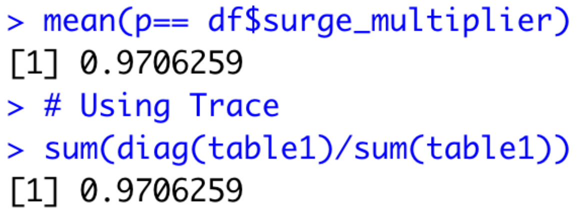
Problem 2: Select one of the techniques (i.e. CA, Cluster Analysis, LDA) and apply it to some aspect of your final project dataset. Or research a new technique that we have not covered and apply it. Each team member should investigate a different aspect of the dataset.

I conducted Linear Discriminant Analysis on the rideshare datatset to predict whether the ride would incur a surge price multiplier or not. I created a dataframe of variables: price, distance, surge_multiplier, temperature, humidity, and windSpeed for model building.

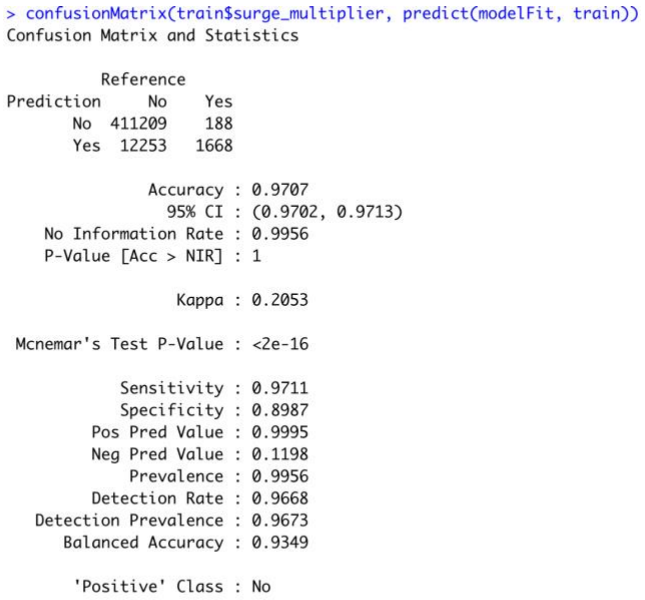


```
p          No     Yes
  No   616733   18472
  Yes     268    2503
```

The model was able to predict 18472 false negatives, and 268 false positives. It had a high accuracy of 97%. The surge multiplier had a value of Yes when the multiplier was greater than 1 and a value of No when the multiplier is equal to 1.



```
> mean(p== df$surge_multiplier)
[1] 0.9706259
> # Using Trace
> sum(diag(table1)/sum(table1))
[1] 0.9706259
```



```
> confusionMatrix(train$surge_multiplier, predict(modelFit, train))
Confusion Matrix and Statistics

          Reference
Prediction     No    Yes
       No  411209    188
       Yes  12253   1668

               Accuracy : 0.9707
                 95% CI : (0.9702, 0.9713)
    No Information Rate : 0.9956
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2053

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.9711
            Specificity : 0.8987
         Pos Pred Value : 0.9995
         Neg Pred Value : 0.1198
             Prevalence : 0.9956
         Detection Rate : 0.9668
   Detection Prevalence : 0.9673
      Balanced Accuracy : 0.9349

       'Positive' Class : No
```

```
library(MASS)
library(DescTools)
library(Hmisc) #Describe Function
library(psych) #Multiple Functions for Statistics and Multivariate Analysis
library(GGally) #ggpairs Function
library(ggplot2) #ggplot2 Functions
library(vioplot) #Violin Plot Function
library(corrplot) #Plot Correlations
library(REdaS) #Bartlett's Test of Sphericity
library(psych) #PCA/FA functions
library(factoextra) #PCA Visualizations
library("FactoMineR") #PCA functions
library(ade4) #PCA Visualizations
library(plyr) # to revalue categorical to factors

'Pre-processing'

#Set working directory
setwd("~/Desktop/Stats Fall 2021")

#Read in dataset
dataset <- read.csv(file="rideshare_kaggle.csv", header=TRUE, sep=",")
```

```
#Check Sample Size and Number of Variables
dim(dataset)
# 693071 sample size and 57 variables

#Check Missing Data
sum(is.na(dataset))
# 55095 missing values.

dataset <- na.omit(dataset)
sum(is.na(dataset))


sum(is.na(dataset))
# 0 missing values

head(dataset)

str(dataset, list.len=ncol(dataset))
str(dataset)
colnames(dataset)

# String variables: icon, short_summary, long_summary, name, product_id, cab_type,
destination
# source, timezone, datetime, id

# Unimportant variables: apparentTemperatureHighTime, timestamp,latitude, longitude,
apparentTemperature
# windGustTime, temperatureHigh, temperatureHighTime, precipIntensity

'Important variables: timestamp, price, distance, surge_multiplier, temperature, humidity
windSpeed, '

# Convert Surge Price to categorical
dataset$surge_multiplier = as.factor(dataset$surge_multiplier)
dataset$surge_multiplier <- recode_factor(dataset$surge_multiplier, "1"="No",
                              "1.25"="Yes", "1.5"="Yes", "1.75"="Yes",
                              "2"="Yes", "2.5"="Yes", "3"="Yes")

table(dataset$surge_multiplier)


View(dataset$surge_multiplier)
```

```
# Subset numerical values
num_data <- subset(dataset, select=-c(icon, short_summary, long_summary, name,
                              product_id, cab_type, destination, source, timezone, datetime, id))
table(num_data)

head(num_data)

# Important variables

df <- subset(num_data, select= c(price, distance, surge_multiplier, temperature, humidity,
                          windSpeed))
head(df)



table(df$surge_multiplier)

library(caret)
library(rpart)
library(tidyverse)
library(dplyr)
library(caret)

# With Cross Validation
# The dependent variable must be categorical
SurgeLevelLDA <- lda(surge_multiplier ~ ., data = df)
SurgeLevelLDA

# Try to predict the class from the original data
p <- predict(SurgeLevelLDA, newdata=df)$class
p

df$SurgeLevel

# Compare the results of the prediction (Confusion Matrix)
table1 <- table(p,df$surge_multiplier)
table1

# Using Trace
sum(diag(table1)/sum(table1))
mean(p== df$surge_multiplier)
#0.91
```

```
#Creating Training and Testing Samples
require(caTools)  # loading caTools library
library(caTools)
set.seed(123)   #  set seed to ensure you always have same random numbers generated
sample = sample.split(df,SplitRatio = 0.70) # splits the data in the ratio mentioned in SplitRatio.
After splitting marks these rows as logical TRUE and the the remaining are marked as logical
FALSE
train =subset(df,sample ==TRUE) # creates a training dataset named train1 with rows which are
marked as TRUE
test=subset(df, sample==FALSE)

# The dependent variable must be categorical (Assuming No Cross-Validation)
surge_multiplierLDA1 = lda(surge_multiplier ~ ., data=train)
surge_multiplierLDA1

prd<-predict(surge_multiplierLDA1, train)

# Compare the results of the prediction
table2 <- table(prd, train$surge_multiplier)

library(caret)
modelFit<- train(surge_multiplier ~ ., method='lda',preProcess=c('scale', 'center'), data=train)

#Confusion Matrix
confusionMatrix(train$surge_multiplier, predict(modelFit, train))
```

Problem 3: Using Google Scholar, locate a journal article, which uses cluster analysis in your field of interest. Write a summary of the journal article and how it utilizes the cluster analysis in two to three paragraphs. Cite the paper in APA format.

Evaluating the determinants of household electricity consumption using cluster analysis

This article aims to identify determinants of household electricity use as a means to promote efficient energy use. The article uses k-means clustering to find “four distinct groups distinguishable by dwelling type, tenure, the number of rooms, the number of bedrooms, annualized electricity consumption and the number of appliances” (Ofetotse, 2021). The clusters were able to provide insight of the baseline factors affecting electricity consumption characteristics of different households. This cluster analysis method developed using questionnaire data of 310 households and a feature selection procedure that maximizes the silhouette was used to select the variables with the most significant clustering tendency. For this research, the squared Euclidean distance was used.

Cluster evaluation was carried out through the use of cluster validity indices (CVIs) such as Davies-Bouldin (DB), Calinski-Harabasz (CH)and Silhouette (SIL). Of the three indices, silhouette performed the best and was selected as the CVI for the study. The average silhouette was used to measure how tightly grouped all the data in the cluster were and compared the results for the different number of clusters.

The study shows that cluster 3 and 4 had higher light energy consumption than entertainment while for cluster 1 and 2 the reverse was true. This was due to the high ownership and use of lighting appliances such as compact fluorescent lights (CFLs) in cluster 3 and 4 and high ownership and use of entertainment appliances such as television sets, video players and game consoles in cluster 1 and 2. The analysis suggests that energy consumers are characterized by dwelling attributes to include dwelling type, tenure and building size, as well as appliance, attributes to include ownership, the type, the number and intensity of use.

Ofetotse, E. L., Essah, E. A., & Yao, R. (2021). Evaluating the determinants of household electricity consumption using cluster analysis. *Journal of Building Engineering*, *43*, 102487. https://doi.org/10.1016/j.jobe.2021.102487

An academic paper from a conference or Journal will be posted to the Homework 4 content section of D2L. Review the paper and evaluate their usage of FA and LDA. In particular address the following: **(See article on Face Recognition using Principle Component Analysis and Linear Discriminant Analysis)**

What is the application of this paper?

This paper compares Principal Component Analysis and Linear Discriminant Analysis as techniques for facial recognition analysis. The study compares the performance and accuracy of PCA and LDA in the application of facial recognition based on various sample sizes using images from UMIST and ORL databases.

What is the research questioning the authors wish to answer in this paper?

The author wishes to discuss which technique, PCA or LDA is more accurate given different sample sizes and image properties. The author attempts to find out which algorithm is best suited based on the sample size and image database used.

What is Face Recognition and what can we learn from it?

Facial recognition is a form of pattern recognition where human visual perception is imitated in a computer. It is the way a computer or system can identify a person through a image of their face through pictures, films, or in real-time. This technique is used in security systems.

How does this paper utilize PCA and LDA in Face Recognition?

UMIST and ORL databases are used to collect facial images. The paper uses LDA by classifying images into classes and calculating the mean of the faces by calculating the center of gravity of sample images. Next, number of samples within classes is found to calculate the ratio of the scatter between matrix and scatter within matrix. Next, eigen values and eigen vectors are found and normalized. Finally the weight and Euclidian distance is found which is able to distinguish and identify faces.

For PCA, first mean faces and standard deviation is calculated based on the center of gravity of sample images. Variance and covariance is found and the covariance matrix is created. Eigenvalues and eigen vectors are calculated. Eigen faces are used to identify the directions in which the main images are different from the mean images. After computing eigenfaces, the weight of the eigenfaces of training images are calculated. After weight calculation, all the above process will be computed for test images. To calculate the weight of the test image, the Euclidian distance between the weight of the train images and test images are calculated. The image whose Euclidian distance will be smaller, that image will be recognized as the correct image.

- What are the results and conclusions from this paper?

Based on the results of the trials, that LDA has a higher average accuracy than PCA for both the UMIST and ORL databases. However, LDA suffers from poor accuracy with data that has large dimensions and limited sample size. The eigenfaces in PCA can effectively compare test and learned training images making PCA a viable option for facial recognition.

- What other areas or fields do you think would benefit from LDA?

LDA can be used to find a linear combination of features that distinguishes two or more object or event classes. LDA can be used in bankruptcy prediction to explain which firms entered bankruptcy vs survived. It can be used in biomedical studies for prognosis of disease outcome or to classify and define groups of different biological objects like types of salmonella. Predicting issues such as bankruptcy or whether it will rain or not is difficult.

- What other thoughts do you have on topic modeling, Face Recognition, and LDA?

Both PCA and LDA are good method for facial recognition. Compared to PCA, LDA deals directly with discrimination between lasses whereas PCA deals with the data in its entirely without taking into consideration class structure. Based on the database and sample size, the appropriate technique should be used.