Megan

Ran Zhang

1/24/2022

Load packages and set working directory

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✓ ggplot2 3.3.5     ✓ purrr   0.3.4
## ✓ tibble  3.1.6     ✓ dplyr   1.0.7
## ✓ tidyr   1.1.4     ✓ stringr 1.4.0
## ✓ readr   2.1.0     ✓ forcats 0.5.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readxl)
library(lattice)

setwd("~/Documents/Private/Work/Nemours/2022/Megan")
```

Load the dataset and manipulate

```
# Read the excel file
df <- read_excel('df.xlsx')

# Convert to tibble for manipulation
df <- as_tibble(df)

# Convert treatment to factor
df_new <- df %>% mutate(treatment = ifelse(treatment == "No Meds", 0, 1))
df_new$treatment <- as.factor(df_new$treatment)

# Pivot-longer
df_long <- df_new %>% pivot_longer(cols=c('MIG_CXCL9', 'IP10_CXCL10', 'Eotaxin2_CCL24'),
names_to = 'type')
```

Box Plot ******************

Boxplot

```
# MIG_CXCL9
MIG_CXCL9_boxplot <- ggplot(df, aes(x=treatment, y=MIG_CXCL9, fill=treatment)) + geom_bo
xplot(alpha=0.7)  + theme_classic() + theme(legend.position="none")
# ggsave('MIG_CXCL9_boxplot.jpg', MIG_CXCL9_boxplot, dpi = 600)

# IP10_CXCL10
IP10_CXCL10_boxplot <- ggplot(df, aes(x=treatment, y=IP10_CXCL10, fill=treatment)) + geo
m_boxplot(alpha=0.7)  + theme_classic() + theme(legend.position="none")
# ggsave('IP10_CXCL10_boxplot.jpg', IP10_CXCL10_boxplot, dpi = 600)

# Eotaxin2_CCL24
Eotaxin2_CCL24_boxplot <- ggplot(df, aes(x=treatment, y=Eotaxin2_CCL24, fill=treatment))
+ geom_boxplot(alpha=0.7)  + theme_classic() + theme(legend.position="none")
# ggsave('Eotaxin2_CCL24_boxplot.jpg', Eotaxin2_CCL24_boxplot, dpi = 600)
```

Scatter Pair Plot ******************

```
library("ggplot2")
library("GGally")
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(tidyverse)
```

```
df_pair <- df %>% select(MIG_CXCL9, IP10_CXCL10, Eotaxin2_CCL24)
pair_plot <- ggpairs(data = df_pair)
# ggsave('pair_plot.jpg', pair_plot, dpi = 600)
```

Linear Discriminant Analysis ****************************

Linear Discriminant Analysis

```
library(caret)
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
theme_set(theme_classic())
```

Split the data

```
set.seed(666)
train.index <- df_new$treatment %>% createDataPartition(p=0.7, list=FALSE)
df.train <- df_new[train.index, ]
df.test <- df_new[-train.index, ]
```

LDA on individual variable

```
library(MASS) # This package will supress the dplyr package select function
```

```
##
## Attaching package: 'MASS'
```

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

```
library(caret) # For confusion matrix
library(MLmetrics) # For f1 score
```

```
##
## Attaching package: 'MLmetrics'
```

```
## The following objects are masked from 'package:caret':
##
##     MAE, RMSE
```

```
## The following object is masked from 'package:base':
##
##     Recall
```

MIG_CXCL9

```
# Build the model
MIG_CXCL9.model <- lda(treatment ~ MIG_CXCL9, data = df.train)

# Make the prediction
MIG_CXCL9.prediction <- MIG_CXCL9.model %>% predict(df.test)

# Confustion matrix
print(confusionMatrix(MIG_CXCL9.prediction$class, df.test$treatment))
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction 0 1
##          0 0 0
##          1 4 7
## 
##                Accuracy : 0.6364
##                  95% CI : (0.3079, 0.8907)
##     No Information Rate : 0.6364
##     P-Value [Acc > NIR] : 0.6322
## 
##                   Kappa : 0
## 
##  Mcnemar's Test P-Value : 0.1336
## 
##             Sensitivity : 0.0000
##             Specificity : 1.0000
##          Pos Pred Value :    NaN
##          Neg Pred Value : 0.6364
##              Prevalence : 0.3636
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
##       Balanced Accuracy : 0.5000
## 
##        'Positive' Class : 0
## 
```

```
# F1 score
print(F1_Score(df.test$treatment, MIG_CXCL9.prediction$class))
```

```
## [1] NaN
```

IP10_CXCL10

```
# Build the model
IP10_CXCL10.model <- lda(treatment ~ IP10_CXCL10, data = df.train)

# Make the prediction
IP10_CXCL10.prediction <- IP10_CXCL10.model %>% predict(df.test)

# Confustion matrix
print(confusionMatrix(IP10_CXCL10.prediction$class, df.test$treatment))
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction 0 1
##          0 2 3
##          1 2 4
##                                           
##                Accuracy : 0.5455          
##                  95% CI : (0.2338, 0.8325)
##     No Information Rate : 0.6364          
##     P-Value [Acc > NIR] : 0.8273          
##                                           
##                   Kappa : 0.0678          
##                                           
##  Mcnemar's Test P-Value : 1.0000          
##                                           
##             Sensitivity : 0.5000          
##             Specificity : 0.5714          
##          Pos Pred Value : 0.4000          
##          Neg Pred Value : 0.6667          
##              Prevalence : 0.3636          
##          Detection Rate : 0.1818          
##    Detection Prevalence : 0.4545          
##       Balanced Accuracy : 0.5357          
##                                           
##        'Positive' Class : 0               
## 
```

```
# F1 score
print(F1_Score(df.test$treatment, IP10_CXCL10.prediction$class))
```

```
## [1] 0.4444444
```

Eotaxin2_CCL24

```
# Build the model
Eotaxin2_CCL24.model <- lda(treatment ~ Eotaxin2_CCL24, data = df.train)

# Make the prediction
Eotaxin2_CCL24.prediction <- Eotaxin2_CCL24.model %>% predict(df.test)

# Confustion matrix
print(confusionMatrix(Eotaxin2_CCL24.prediction$class, df.test$treatment))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 2 0
##          1 2 7
##
##                Accuracy : 0.8182
##                  95% CI : (0.4822, 0.9772)
##     No Information Rate : 0.6364
##     P-Value [Acc > NIR] : 0.1750
##
##                   Kappa : 0.56
##
##  Mcnemar's Test P-Value : 0.4795
##
##             Sensitivity : 0.5000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.7778
##              Prevalence : 0.3636
##          Detection Rate : 0.1818
##    Detection Prevalence : 0.1818
##       Balanced Accuracy : 0.7500
##
##        'Positive' Class : 0
##
```

```
# F1 score
print(F1_Score(df.test$treatment, Eotaxin2_CCL24.prediction$class))
```

```
## [1] 0.6666667
```

Three variables together to build the model

```
df.model <- lda(treatment ~ MIG_CXCL9 + IP10_CXCL10 + Eotaxin2_CCL24, data = df.train)

# Make the prediction
df.prediction <- df.model %>% predict(df.test)

# Confustion matrix
print(confusionMatrix(df.prediction$class, df.test$treatment))
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction 0 1
##          0 2 3
##          1 2 4
##                                           
##                Accuracy : 0.5455          
##                  95% CI : (0.2338, 0.8325)
##     No Information Rate : 0.6364          
##     P-Value [Acc > NIR] : 0.8273          
##                                           
##                   Kappa : 0.0678          
##                                           
##  Mcnemar's Test P-Value : 1.0000          
##                                           
##             Sensitivity : 0.5000          
##             Specificity : 0.5714          
##          Pos Pred Value : 0.4000          
##          Neg Pred Value : 0.6667          
##              Prevalence : 0.3636          
##          Detection Rate : 0.1818          
##    Detection Prevalence : 0.4545          
##       Balanced Accuracy : 0.5357          
##                                           
##        'Positive' Class : 0               
## 
```

```
# F1 score
print(F1_Score(df.test$treatment, df.prediction$class))
```

```
## [1] 0.4444444
```