

PCA vs LDA

Monika Osiak, Anna Pręgowska, Patrycja Szczepaniak, Rafał Szulejko

15 04 2020

Wstęp

Zadaniem było wykonanie analizy porównawczej metody wizualizacji PCA i LDA. Zadanie zostało wykonane na podstawie zbioru danych dotyczącego problemów z kręgosłupem. Zbiór posiada 2 klasy: normal i abnormal.

```
head(spine)
```

```
##      pelvic_incidence pelvic_tilt lumbar_lordosis_angle sacral_slope pelvic_radius
## 1          63.02782    22.552586          39.60912    40.47523      98.67292
## 2          39.05695    10.060991          25.01538    28.99596     114.40543
## 3          68.83202    22.218482          50.09219    46.61354     105.98514
## 4          69.29701    24.652878          44.31124    44.64413     101.86850
## 5          49.71286     9.652075          28.31741    40.06078     108.16872
## 6          40.25020    13.921907          25.12495    26.32829     130.32787
##      degree_spondylolisthesis pelvic_slope direct_tilt thoracic_slope
## 1             -0.254400    0.7445035    12.5661      14.5386
## 2             4.564259    0.4151857    12.8874      17.5323
## 3            -3.530317    0.4748892    26.8343      17.4861
## 4            11.211523    0.3693453    23.5603      12.7074
## 5             7.918501    0.5433605    35.4940      15.9546
## 6             2.230652    0.7899929    29.3230      12.0036
##      cervical_tilt sacrum_angle scoliosis_slope      class      X
## 1          15.30468    -28.658501          43.5123 Abnormal  NA
## 2          16.78486    -25.530607          16.1102 Abnormal  NA
## 3          16.65897    -29.031888          19.2221 Abnormal  NA
## 4          11.42447    -30.470246          18.8329 Abnormal  NA
## 5           8.87237    -16.378376          24.9171 Abnormal  NA
## 6          10.40462     -1.512209           9.6548 Abnormal  NA
```

PCA

Rozkład PCA dokonywany jest za pomocą metody prcomp.

```
spine.pr <- prcomp(spine[c(1:12)] ,
                  center=TRUE,
                  scale=TRUE)
```

Następnie obliczamy wariancję w dwóch pierwszych składnikach wiodących.

```
pc1_var <- as.double(summary(spine.pr)$importance[,1][2])
pc2_var <- as.double(summary(spine.pr)$importance[,2][2])
print(sprintf('Wariancja danych zawarta w pierwszym składniku wiodącym: %s%%',
              format(round(pc1_var, 2), nsmall = 2)))
```

```
## [1] "Wariancja danych zawarta w pierwszym składniku wiodącym: 0.27%"
```

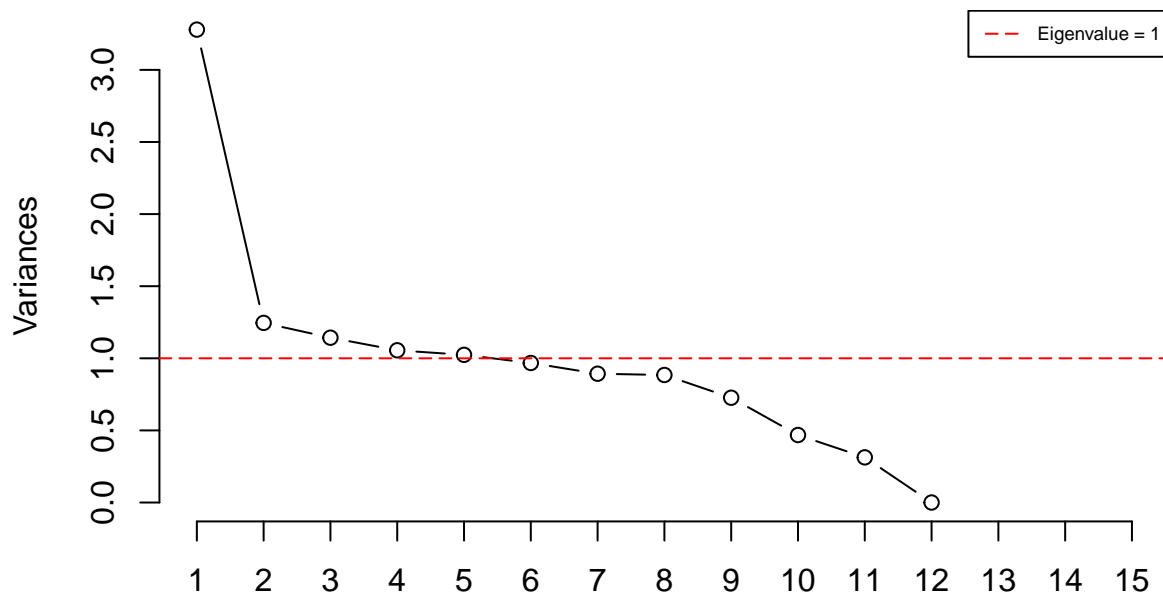
```
print(sprintf('Wariancja danych zawarta w drugim składniku wiodącym: %s%%',
              format(round(pc2_var, 2), nsmall = 2)))
```

```
## [1] "Wariancja danych zawarta w drugim składniku wiodącym: 0.10%"
```

Prezentacja pozostałych składników wiodących:

```
screepplot(spine.pr,
            type="l",
            npcs=15,
            main="Screeplot of the first 10 PCs")
abline(h=1,
       col="red",
       lty=5)
legend("topright",
       legend=c("Eigenvalue = 1"),
       col=c("red"),
       lty=5,
       cex=0.6)
```

Screeplot of the first 10 PCs



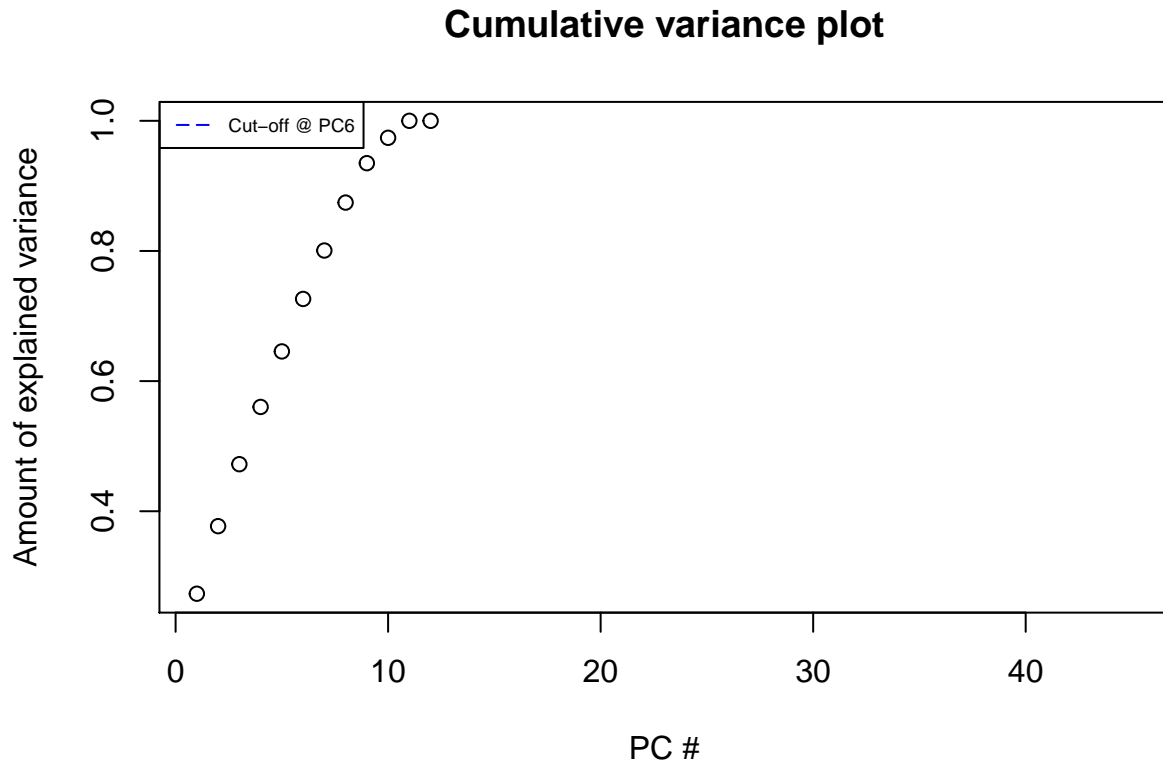
Zgodnie z oczekiwaniami, łączna wariancja kolejnych składników zbliża się do jedności.

```
cumpro <- cumsum(spine.pr$sdev^2 / sum(spine.pr$sdev^2))
plot(cumpro[0:45],
     xlab="PC #",
     ylab="Amount of explained variance",
     main="Cumulative variance plot")
legend("topleft",
```

```

legend=c("Cut-off @ PC6"),
col=c("blue"),
lty=5,
cex=0.6)

```



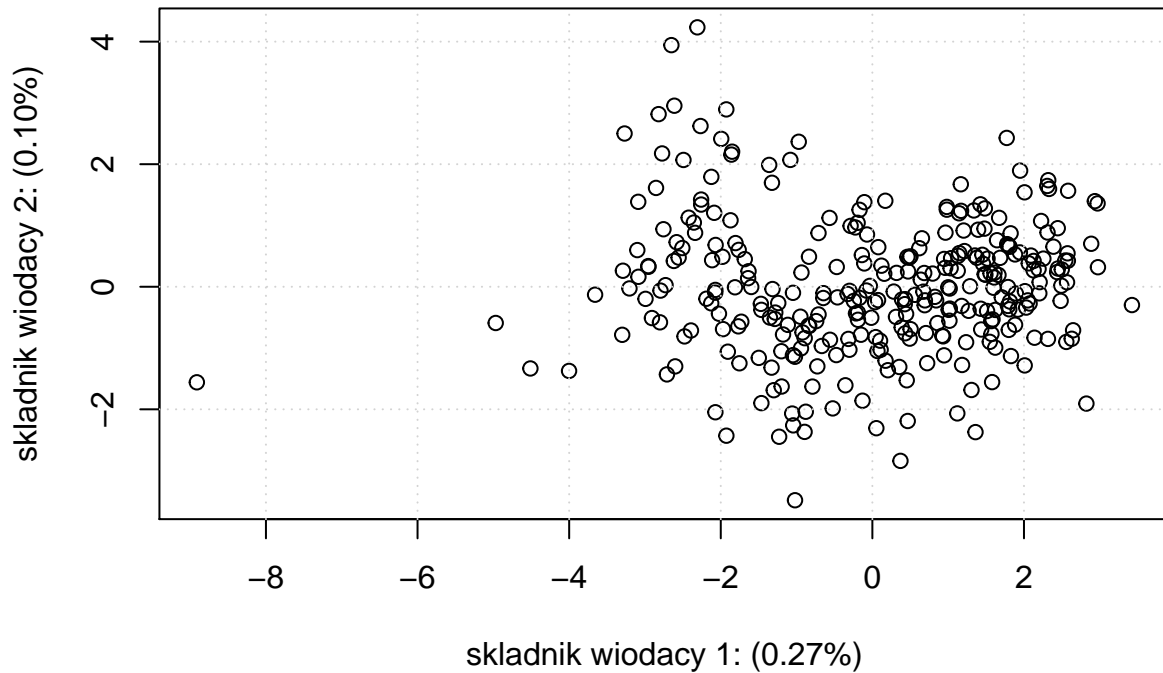
Po zredukowaniu danych do dwóch wymiarów otrzymujemy poniższy wykres:

```

x_label <- sprintf('składnik wiodący 1: (%s%%)', format(round(pc1_var, 2), nsmall = 2))
y_label <- sprintf('składnik wiodący 2: (%s%%)', format(round(pc2_var, 2), nsmall = 2))
plot_title <- 'Dane po redukcji rozmiaru do dwóch wymiarów'
plot(spine.pr$x[,1],
     spine.pr$x[,2],
     xlab=x_label,
     ylab=y_label,
     main=plot_title)
grid(nx=NULL,
     ny=NULL,
     col="lightgray",
     lty="dotted")

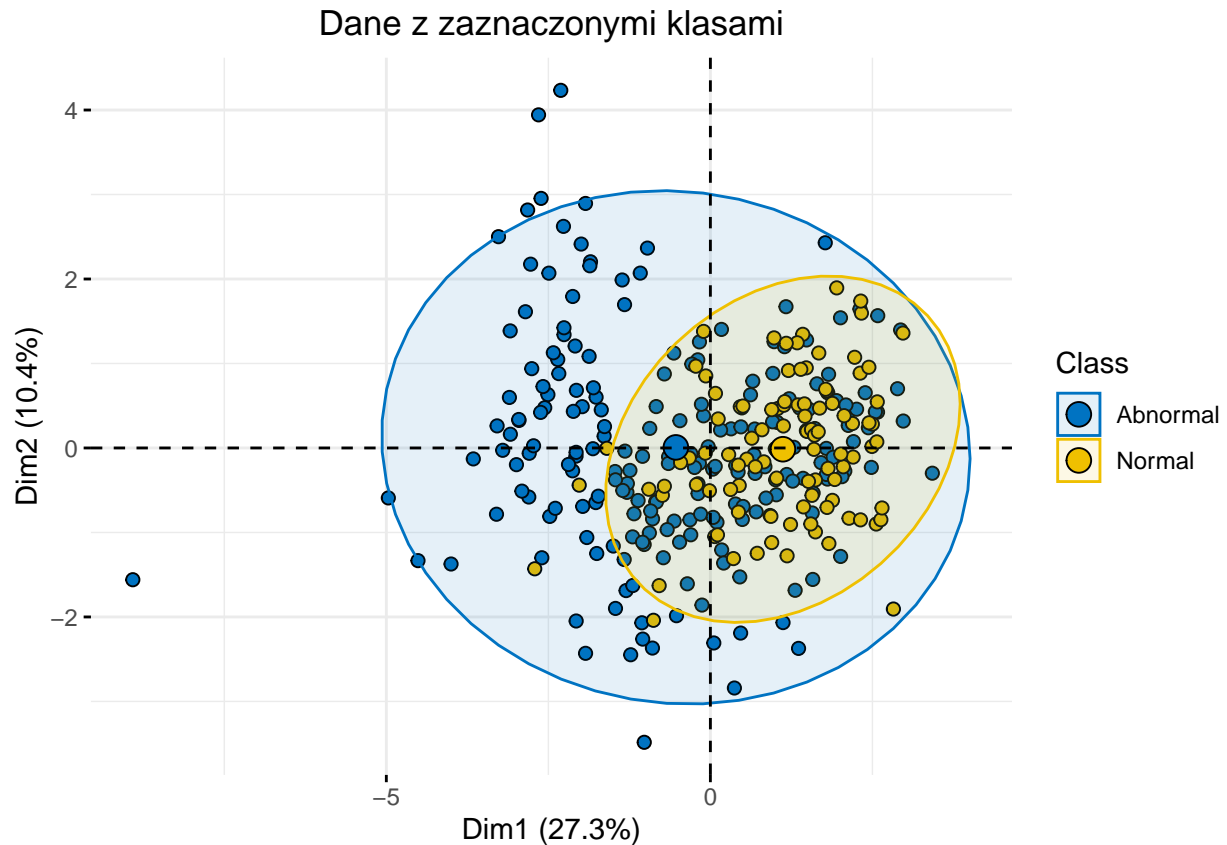
```

Dane po redukcji rozmiaru do dwóch wymiarów



Ostateczny wynik uzyskujemy po podziale na klasy.

```
fviz_pca_ind(spine.pr,  
  geom.ind="point",  
  pointshape=21,  
  pointsize=2,  
  fill.ind=spine$class,  
  col.ind="black",  
  palette="jco",  
  addEllipses=TRUE,  
  label="var",  
  col.var="black",  
  repel=TRUE,  
  legend.title="Class") +  
  ggtitle("Dane z zaznaczonymi klasami") +  
  theme(plot.title=element_text(hjust=0.5))
```



LDA

Rozkład LDA dokonywany jest za pomocą metody `lda`.

Dane zostały już wcześniej wczytane. Dlatego zaczynamy od podziału zbioru na treningowy (80%) i testowy (20%).

```
set.seed(123)
training.samples <- spine$class %>%
  createDataPartition(p = 0.8, list = FALSE)

train.data <- spine[training.samples, ]
test.data <- spine[-training.samples, ]
```

Estymacja parametrów preprocesowania:

```
preproc.param <- train.data %>%
  preProcess(method = c("center", "scale"))
```

Transformacja danych przy użyciu estymowanych parametrów:

```
train.transformed <- preproc.param %>% predict(train.data)
test.transformed <- preproc.param %>% predict(test.data)
```

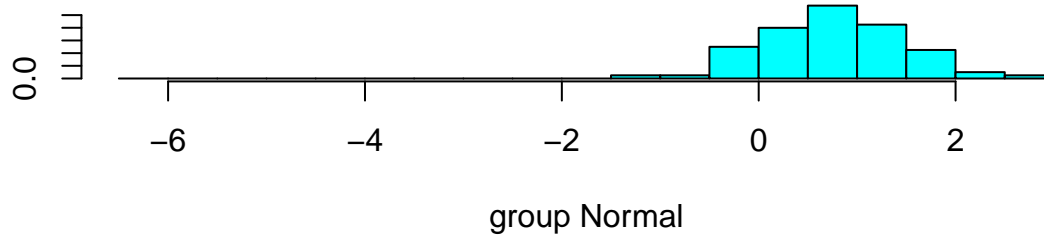
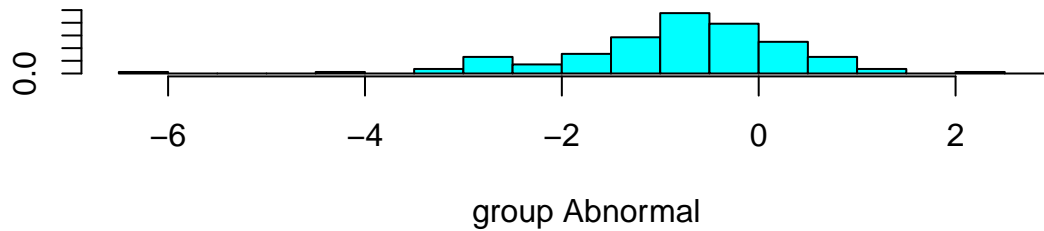
Dopasowanie modelu:

```
model <- lda(formula=class~., data=train.transformed)
```

```
## Warning in lda.default(x, grouping, ...): variables are collinear
```

Prezentacja modelu z podziałem na klasy.

```
plot(model)
```



Wykonanie predykcji:

```
predictions <- model %>% predict(train.transformed)
```

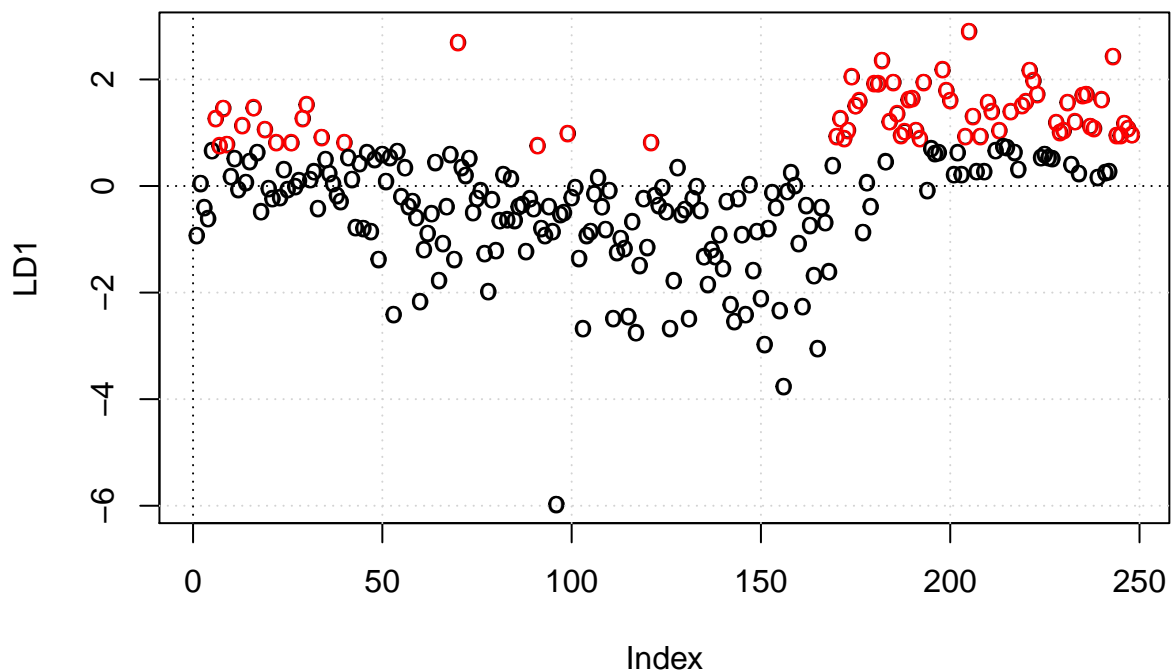
Wyznaczenie precyzji predykcji modelu:

```
mean(predictions$class==test.transformed$class)
```

```
## [1] 0.6330645
```

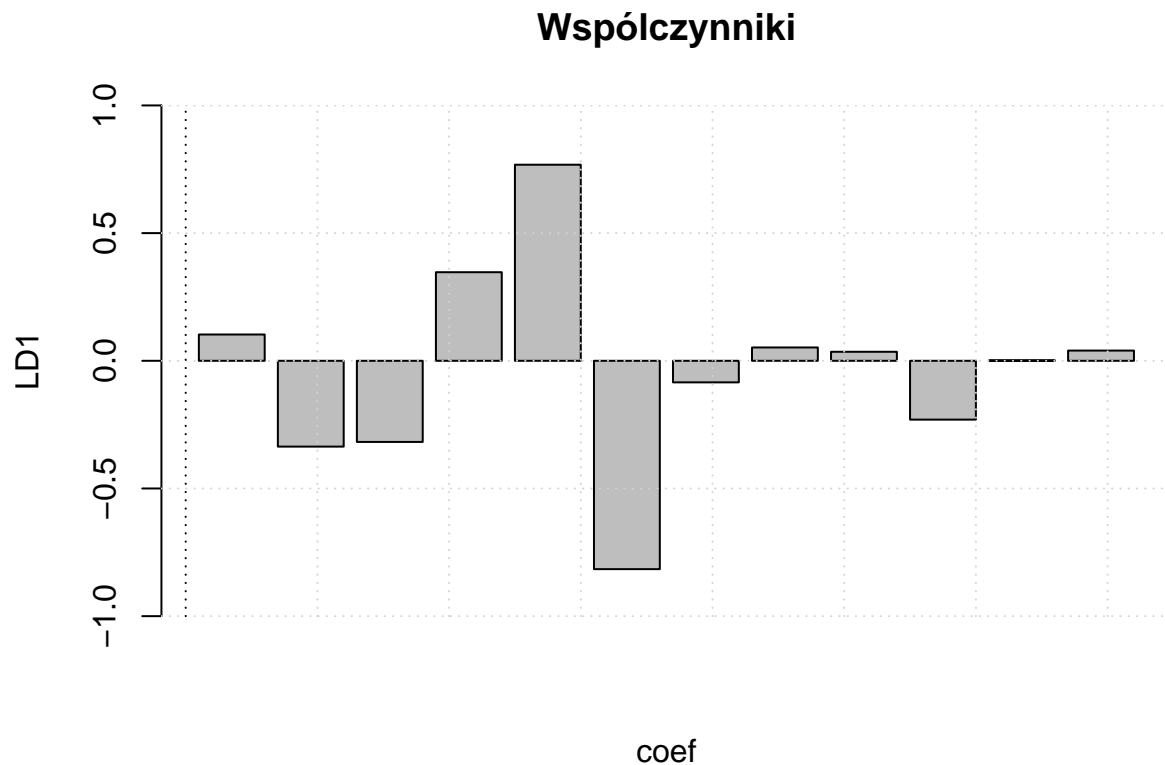
Wizualizacja funkcji dyskryminacji:

```
plot_title <- ''
plot(predictions$x[,1],
      ylab="LD1",
      main=plot_title)
grid(nx=NULL,
      ny=NULL,
      col="lightgray",
      lty="dotted")
text(predictions$x[,1], label="o", col=c(as.numeric(predictions$class)))
abline(v=0, lty="dotted")
abline(h=0, lty="dotted")
```



Wizualizacja współczynników modelu:

```
plot_title <- 'Współczynniki'
barplot(c(coef(model)), main=plot_title, xlab = "coef", ylab = "LD1", ylim=c(-1,1))
grid(nx=NULL,
      ny=NULL,
      col="lightgray",
      lty="dotted")
abline(v=0, lty="dotted", col=c(as.numeric(predictions$class)))
```



Wnioski

1. Podstawową różnicą między omawianymi metodami jest to, że LDA jest metodą nadzorowaną (występuje wstępny podział na klasy), natomiast PCA jest nienadzorowana (nie posiada wstępnego podziału).
2. Algorytm PCA opiera się na jednej macierzy kowariancji, natomiast LDA korzysta z macierzy opisujących zmienność wewnątrzgrupową i międzygrupową.
3. Wybraliśmy niestety zbiór, który okazał się trudny do skutecznej liniowej eliminacji wymiarów, jednak dla naszego modelu LDA lepiej sklasyfikowało dane.
4. LDA zapewnia lepszą separację klas niż PCA. Dzieje się tak dlatego, że PCA skupia się przede wszystkim na śledzeniu wariancji danych, natomiast LDA na wariancjach międzyklasowych.
5. LDA wymaga wcześniejszego preprocesowania danych, przez co jest trudniejsze w implementacji.
6. Dla metody LDA największy wpływ na klasyfikację miały atrybuty `pelvic_radius` oraz `degree_spondylolisthesis`.
7. Przypuszczamy, że w większości sytuacji metoda LDA powinna dawać lepsze wyniki, jednakże wszystko zależy od omawianego przypadku i do każdego należy podejść indywidualnie.