

SPRAWOZDANIE

Zajęcia: Uczenie Maszynowe

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium Nr 3 Data 07.12.2024 Temat: "Uczenie maszynowe w praktyce: analiza skupień" Wariant 11	Szymon Nycz Informatyka II stopień, niestacjonarne, 1 semestr, gr.1b
---	---

1. Polecenie:

Powikłania zawału mięśnia sercowego:

<https://www.kaggle.com/datasets/rafatashrafjoy/myocardial-infarction-complications>

2. Link do repozytorium:

Link: https://github.com/Maciek332/Semestr_1_Nycz/tree/master/UM

3. Opis programu opracowanego

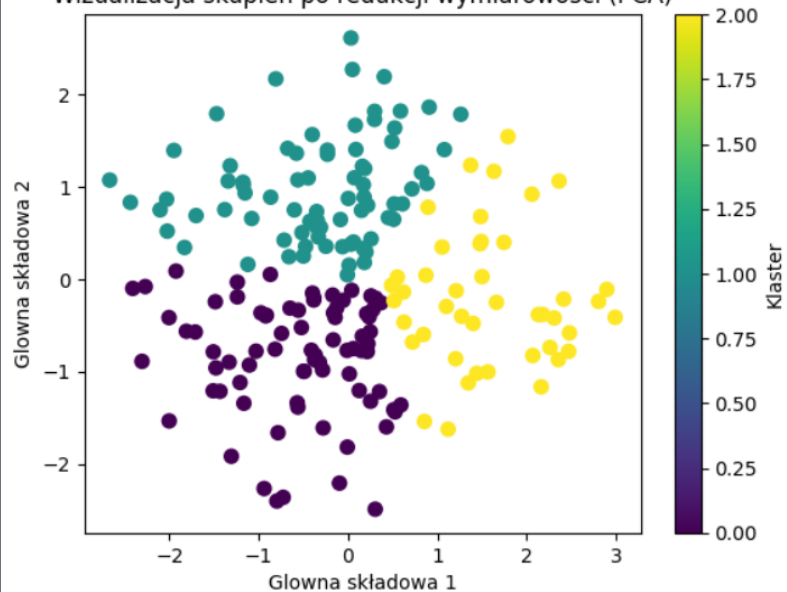
1) Wizualizacja skupień

- Python

```
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Generowanie przykładowych danych
np.random.seed(42)
data = np.random.rand(200, 5) # 200 punktów w 5 wymiarach
# Standaryzacja danych
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
# Redukcja wymiarowości za pomocą PCA
pca = PCA(n_components=2)
data_pca = pca.fit_transform(data_scaled)
# Klasyfikacja k-means (na potrzeby wizualizacji)
kmeans = KMeans(n_clusters=3, random_state=42)
labels = kmeans.fit_predict(data_pca)
# Wizualizacja wyników
plt.scatter(data_pca[:, 0], data_pca[:, 1], c=labels, cmap='viridis', s=50)
plt.xlabel('Główna składowa 1')
plt.ylabel('Główna składowa 2')
plt.title('Wizualizacja skupień po redukcji wymiarowości (PCA)')
plt.colorbar(Label='Klaster')
plt.show()
```

Wizualizacja skupień po redukcji wymiarowości (PCA)



- R

```
# ładowanie bibliotek
library(ggplot2)
library(cluster)

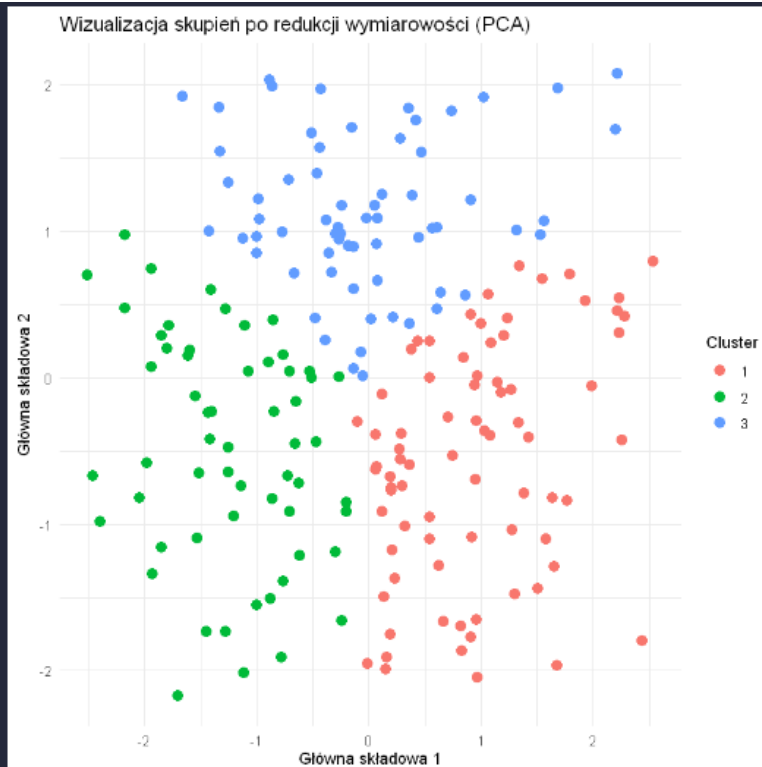
# generowanie przykładowych danych
set.seed(42)
data <- matrix(runif(200 * 5), ncol = 5)

# standaryzacja danych
data_scaled <- scale(data)

# redukcja wymiarowości za pomocą PCA
pca <- prcomp(data_scaled, center = TRUE, scale. = TRUE)
data_pca <- pca$x[, 1:2]

# klasteryzacja k-means (na potrzeby wizualizacji)
kmeans_result <- kmeans(data_pca, centers = 3)

# wizualizacja wyników
data_plot <- data.frame(PC1 = data_pca[, 1], PC2 = data_pca[, 2], cluster = factor(kmeans_result$cluster))
ggplot(data_plot, aes(x = PC1, y = PC2, color = cluster)) +
  geom_point(size = 3) +
  labs(title = "Wizualizacja skupień po redukcji wymiarowości (PCA)",
       x = "Główna składowa 1",
       y = "Główna składowa 2") +
  theme_minimal()
```



2) Klasteryzacja k-means

- Python

```
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

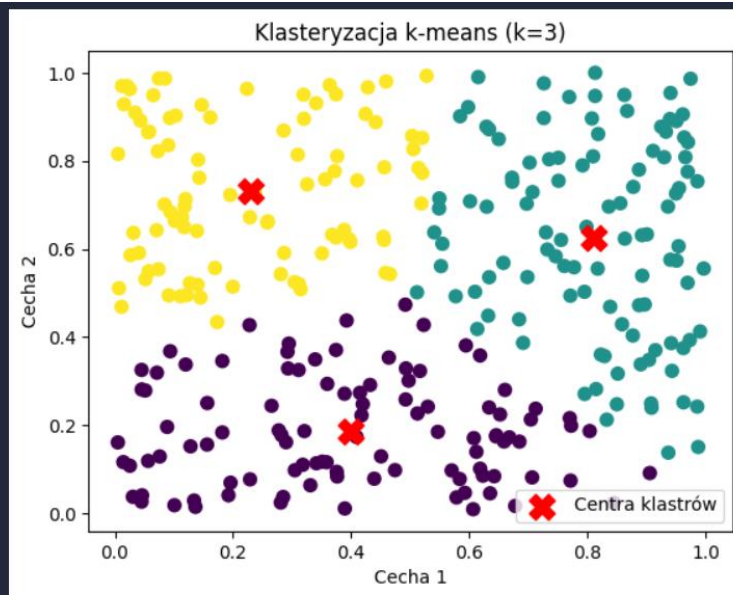
# Generowanie przykładowych danych
np.random.seed(42)
data = np.random.rand(300, 2) # 300 punktów w 2 wymiarach

# Klasteryzacja k-means
k = 3 # liczba klastrów
kmeans = KMeans(n_clusters=k, random_state=42)

labels = kmeans.fit_predict(data)

# Wizualizacja wyników
plt.scatter(data[:, 0], data[:, 1], c=labels, cmap='viridis', s=50)
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
            c='red', marker='X', s=200, label='Centra klastrów')
plt.title(f'Klasteryzacja k-means (k={k})')
plt.xlabel('Cecha 1')
plt.ylabel('Cecha 2')
plt.legend()
plt.show()
```

✓ 0.4s



- R



```
# Ładowanie bibliotek
library(ggplot2)

# Generowanie przykładowych danych
set.seed(42)
data <- data.frame(x = runif(300), y = runif(300))

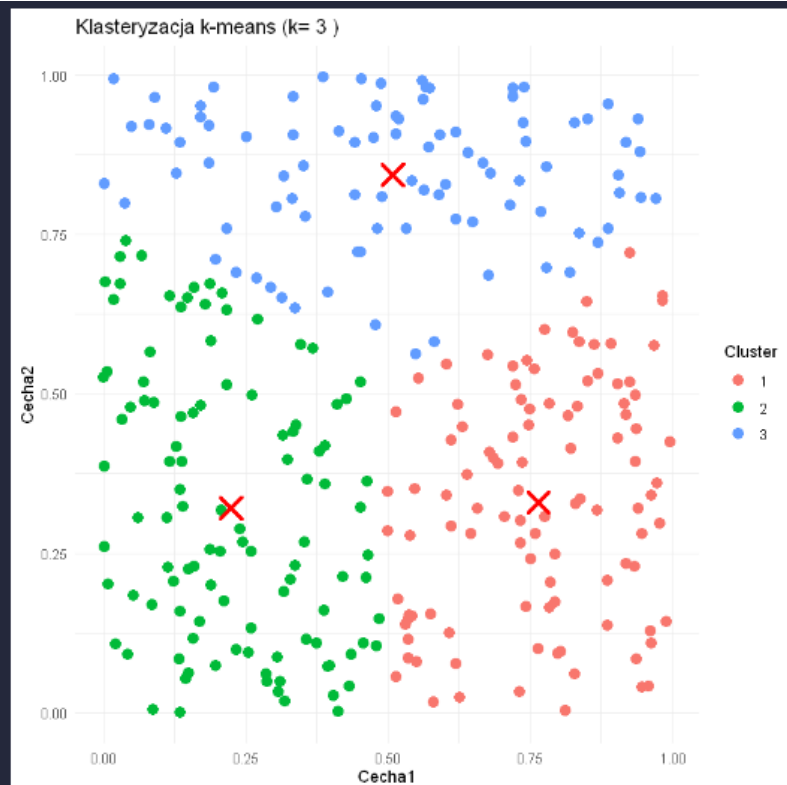
# Klasteryzacja k-means
k <- 3 # liczba klastrow
kmeans_result <- kmeans(data, centers = k)

# Wizualizacja wyników
data$Cluster <- as.factor(kmeans_result$cluster)
centers <- as.data.frame(kmeans_result$centers)
colnames(centers) <- c("x", "y")

ggplot(data, aes(x = x, y = y, color = Cluster)) +
  geom_point(size = 3) +
  geom_point(data = centers, aes(x = x, y = y),
            color = 'red', shape = 4, size = 5, stroke = 2) +
  labs(title = paste("Klasteryzacja k-means (k=", k, ")"),
       x = "Cecha1",
       y = "Cecha2") +
  theme_minimal()
```

[3]

...



3) Klasteryzacja hierarchiczna

- Python

```
import numpy as np
import scipy.cluster.hierarchy as sch
from scipy.cluster.hierarchy import dendrogram, linkage, fcluster
import matplotlib.pyplot as plt

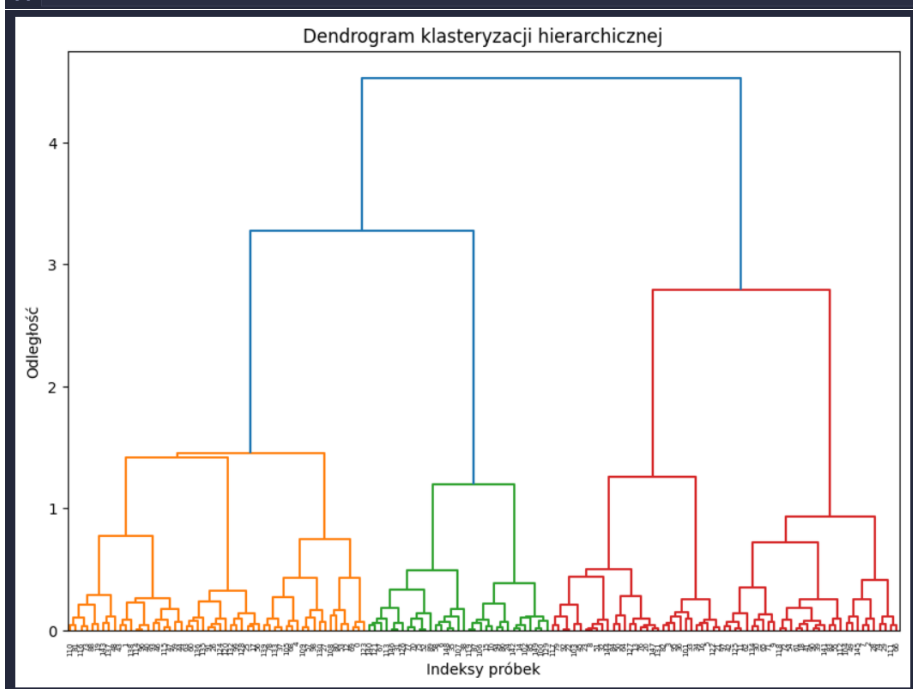
# Generowanie przykładowych danych
np.random.seed(42)
data = np.random.rand(150, 2) # 150 punktów w 2 wymiarach

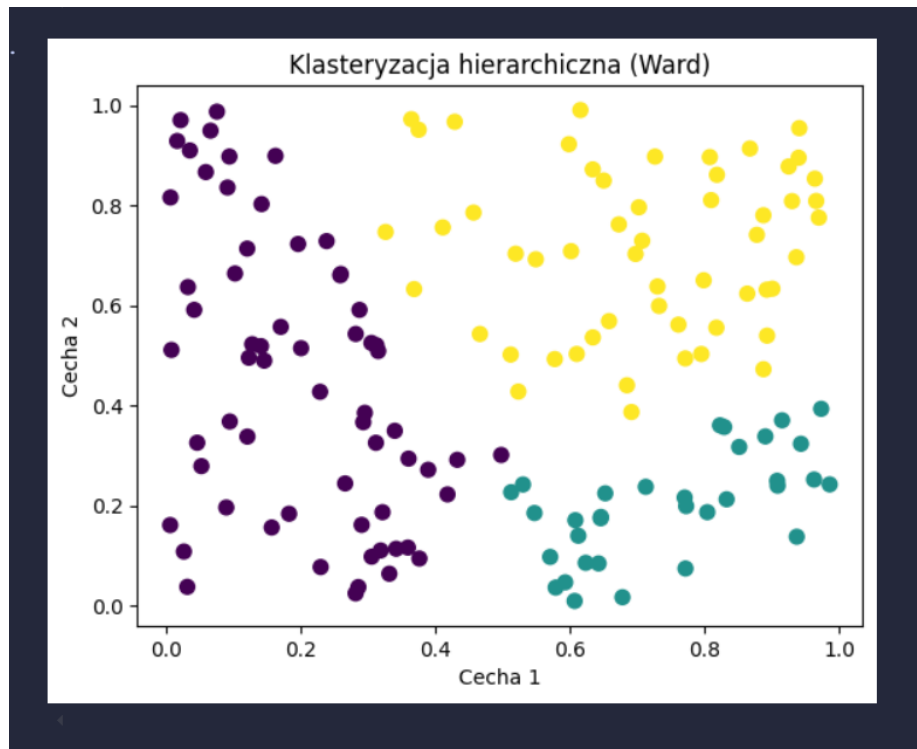
# Klasteryzacja hierarchiczna
linked = linkage(data, method='ward') # Użycie metody Ward

# Wizualizacja dendrogramu
plt.figure(figsize=(10, 7))
dendrogram(linked, orientation='top', distance_sort='descending', show_leaf_counts=True)
plt.title('Dendrogram klasteryzacji hierarchicznej')
plt.xlabel('Indeksy próbek')
plt.ylabel('Odległość')
plt.show()

# Wyodrębnienie klastrów na wybranym poziomie
clusters = fcluster(linked, t=3, criterion='maxclust')

# Wizualizacja klastrów
plt.scatter(data[:, 0], data[:, 1], c=clusters, cmap='viridis', s=50)
plt.title('Klasteryzacja hierarchiczna (Ward)')
plt.xlabel('Cecha 1')
plt.ylabel('Cecha 2')
plt.show()
```





- R

```
# ładowanie bibliotek
library(ggplot2)

# Generowanie przykładowych danych
set.seed(42)
data <- data.frame(x = runif(150), y = runif(150))

# Obliczenie macierzy odległości i klasteryzacja hierarchiczna
distance_matrix <- dist(data)
hc <- hclust(distance_matrix, method = "ward.D2")

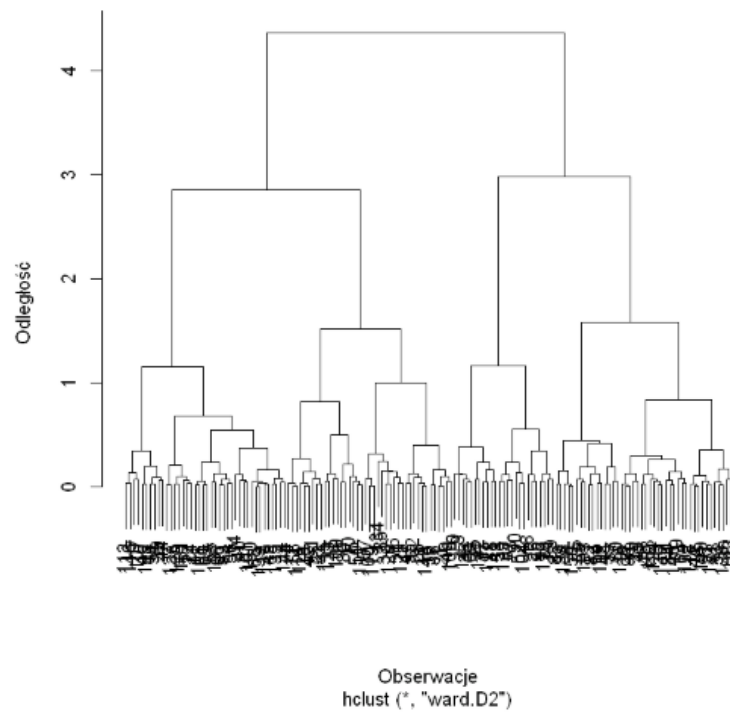
# Wizualizacja dendrogramu
plot(hc, main = "Dendrogram klasteryzacji hierarchicznej", xlab = "Obserwacje", ylab = "Odległość")

# Wyodrębnienie klastrow
clusters <- cutree(hc, k = 3)
data$Cluster <- as.factor(clusters)

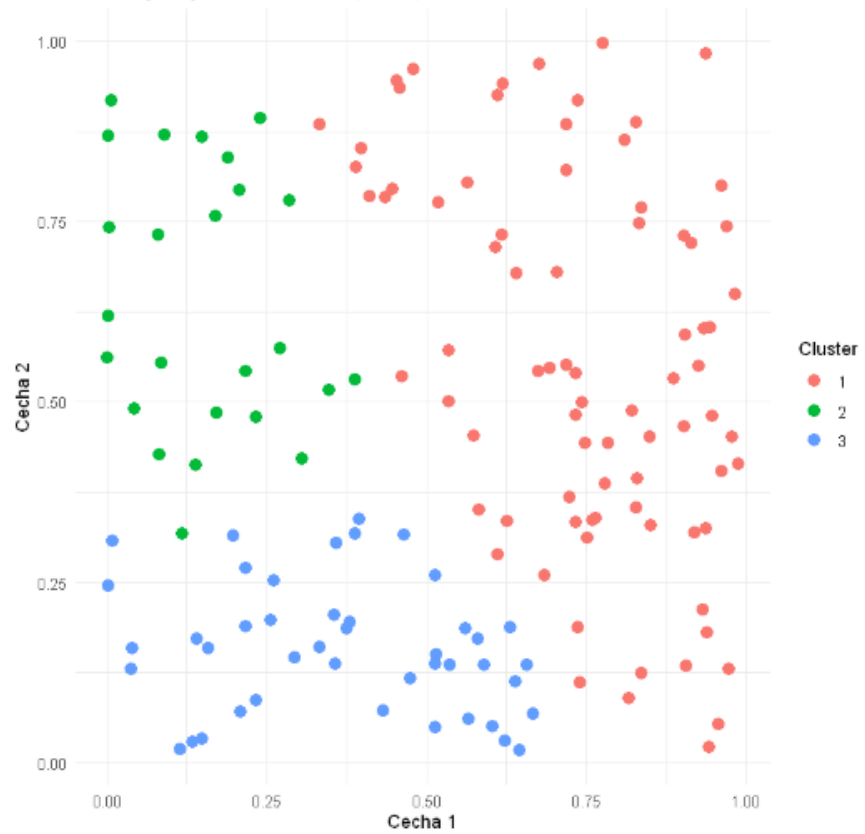
# Wizualizacja skupień
ggplot(data, aes(x = x, y = y, color = Cluster)) +
  geom_point(size = 3) +
  labs(title = "Klasteryzacja hierarchiczna (Ward)",
       x = "Cecha 1",
       y = "Cecha 2") +
  theme_minimal()
```

[4]

Dendrogram klasteryzacji hierarchicznej



Obserwacje
hclust ("ward.D2")
Klasteryzacja hierarchiczna (Ward)



4. Wnioski

Uczenie nienadzorowane to kluczowy typ uczenia maszynowego, którego celem jest identyfikacja ukrytych wzorców w danych bez potrzeby posiadania etykiet (czyli zmiennych wynikowych). Analiza skupień (ang. clustering) polega na grupowaniu danych w k klastrów, w których punkty są bardziej podobne do siebie niż do punktów z innych klastrów. Celem klasteryzacji jest minimalizacja funkcji kosztu $J(C)$, która ocenia wewnętrzną różnorodność w klastrach. Ocena wyników klasteryzacji w uczeniu nienadzorowanym jest trudniejsza niż w uczeniu nadzorowanym, ponieważ brakuje etykiet referencyjnych. Analiza głównych składowych (ang. Principal Component Analysis, PCA) to metoda redukcji wymiarowości danych. PCA przekształca dane wielowymiarowe w mniejszą liczbę wymiarów, zachowując jak najwięcej informacji. Metody niehierarchiczne do analizy skupień, takie jak k -means, dzielą dane na k skupień, gdzie k jest wybranym przez użytkownika parametrem. Metody hierarchiczne tworzą hierarchię klastrów, co pozwala zrozumieć strukturę danych na różnych poziomach szczegółowości. Metryka odległości (np. odległość euklidesowa, Manhattan, kosinusowa) definiuje sposób mierzenia odległości między punktami, podczas gdy sposób łączenia (np. single linkage, complete linkage, average linkage) określa, jak mierzyć odległość między klastrami.