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 10	Anna Więzik Informatyka II stopień, niestacjonarne, 1 semestr, gr.1b

1. Polecenie:

   Smoking patient: https://www.kaggle.com/datasets/thomaskonstantin/cpg-values-of-smoking-and-non-smoking-patients

2. Link do repozytorium:

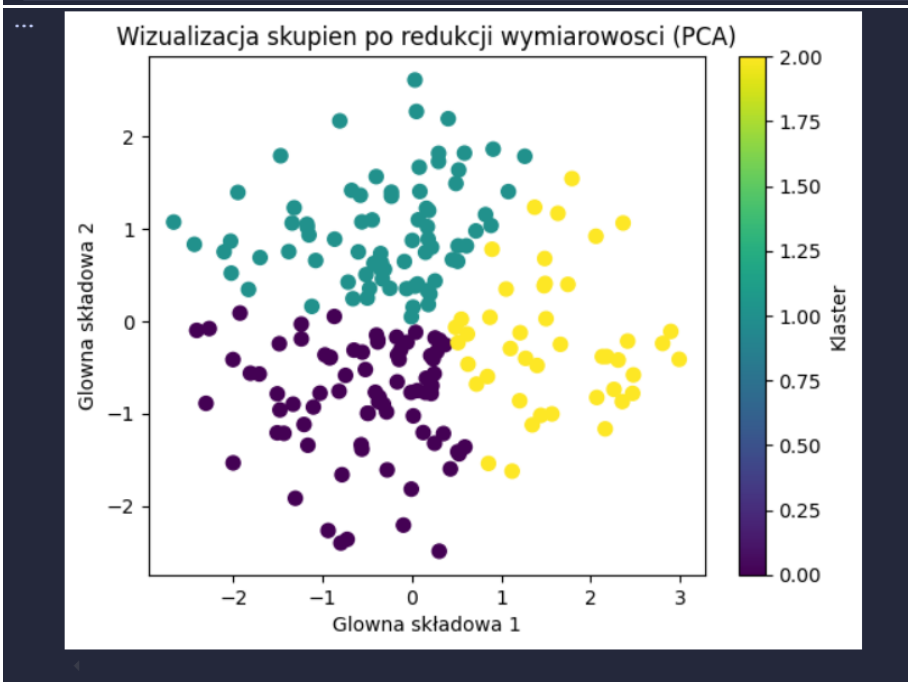
   Link: https://github.com/AnaShiro/UM_2024

3. Opis programu opracowanego

1) Wizualizacja skupień

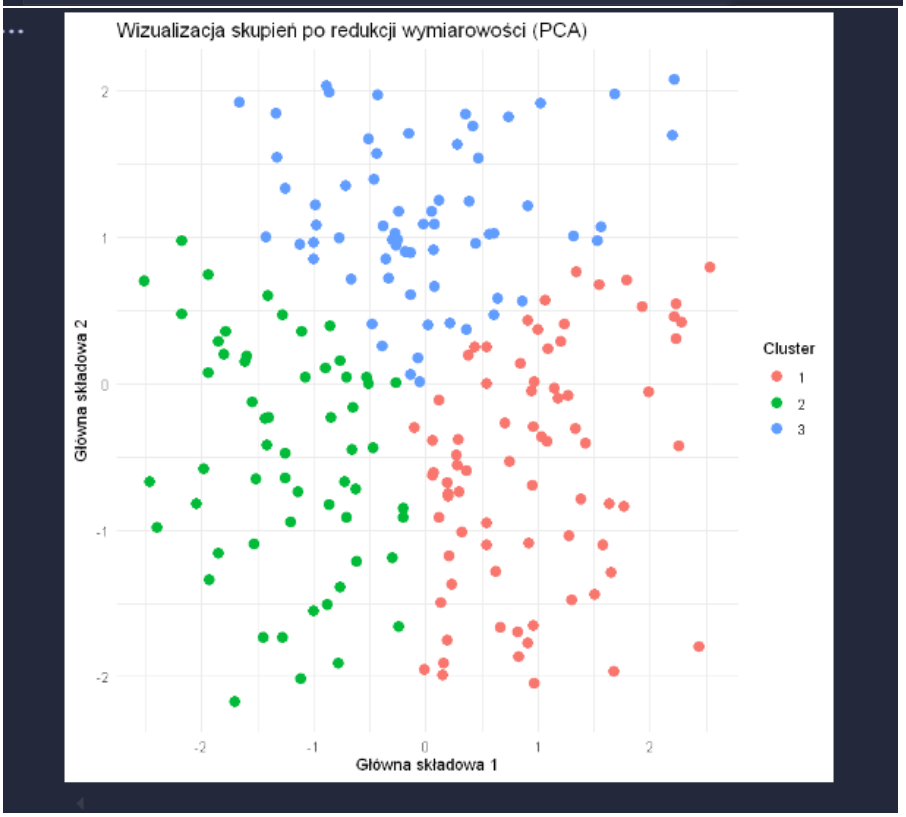
- Python

```
+ Code  + Markdown

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 punktow w 5 wymiarach
# Standaryzacja danych
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
# Redukcja wymiarowosci za pomoca PCA
pca = PCA(n_components=2)
data_pca = pca.fit_transform(data_scaled)
# Klasteryzacja k-means (na potrzeby wizualizacji)
kmeans = KMeans(n_clusters=3, random_state=42)
labels = kmeans.fit_predict(data_pca)
# Wizualizacja wynikow
plt.scatter(data_pca[:, 0], data_pca[:, 1], c=labels, cmap='viridis', s=50)
plt.xlabel('Glowna składowa 1')
plt.ylabel('Glowna składowa 2')
plt.title('Wizualizacja skupien po redukcji wymiarowosci (PCA)')
plt.colorbar(label='Klaster')
plt.show()
1]  ✓ 27.2s
```



- R

```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()
```

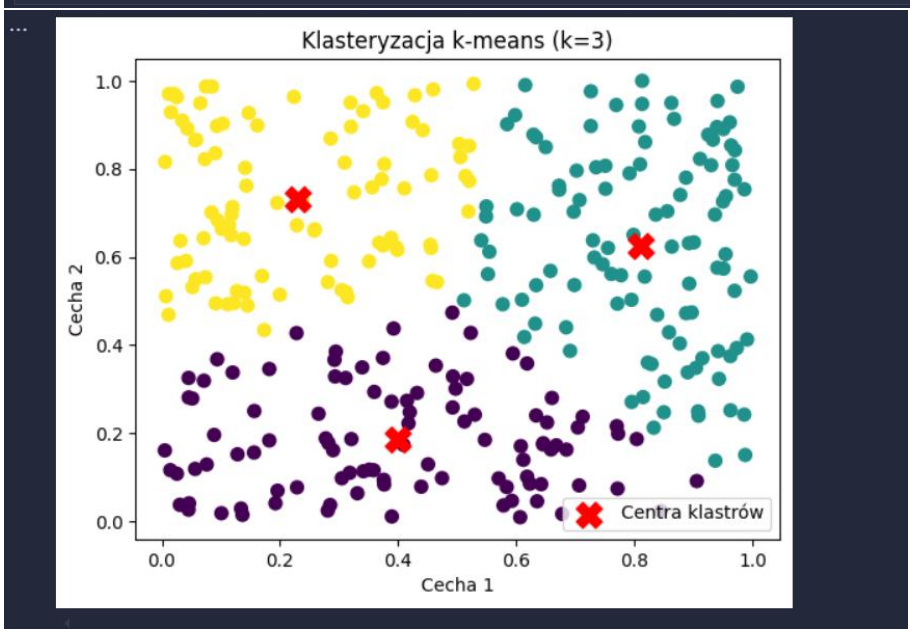


2) Klasteryzacja k-means

- Python

```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()
```

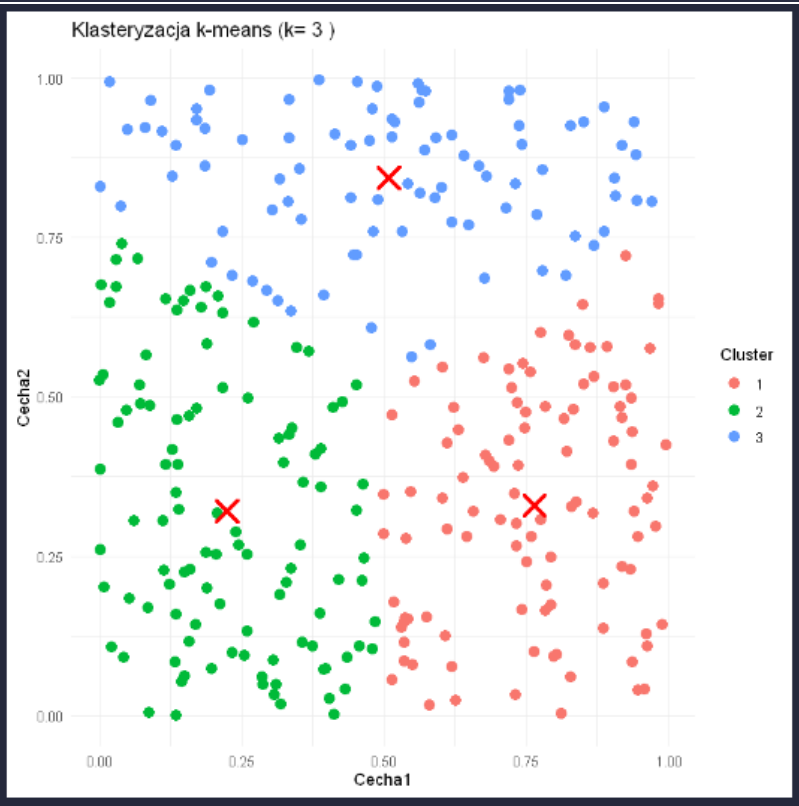
✓ 0.4s



- R

```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 wynikow
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]

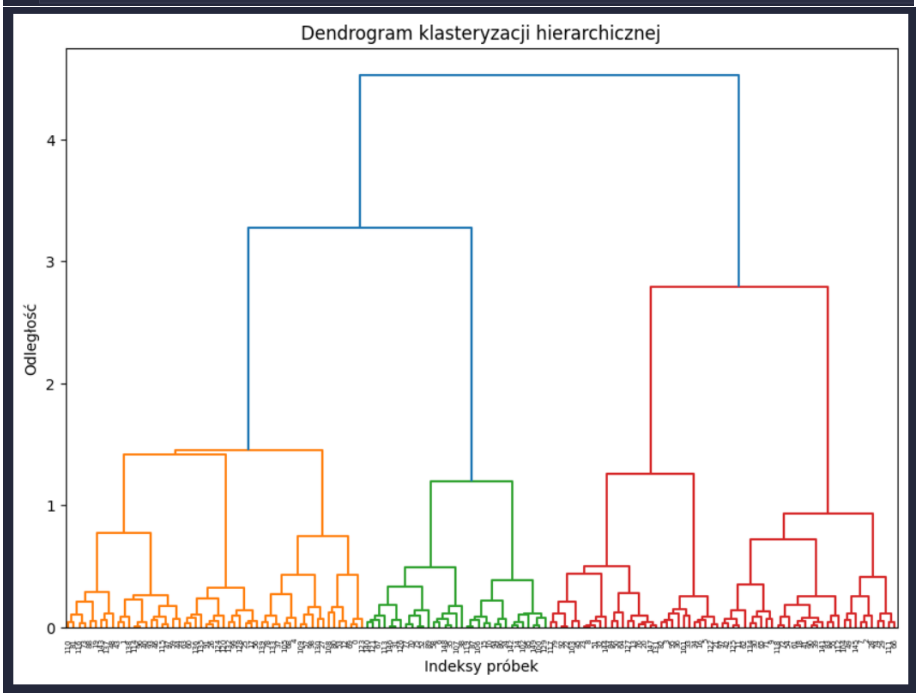


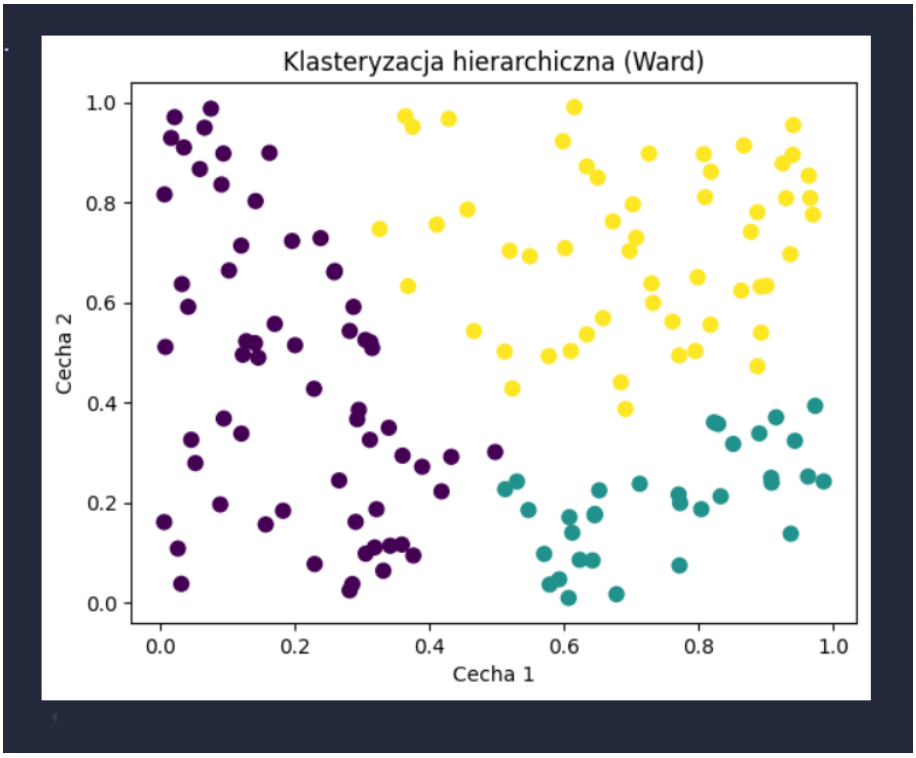
3) Klasteryzacja hierarchiczna

- Python

```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()
```

[3] ✓ 3.3s

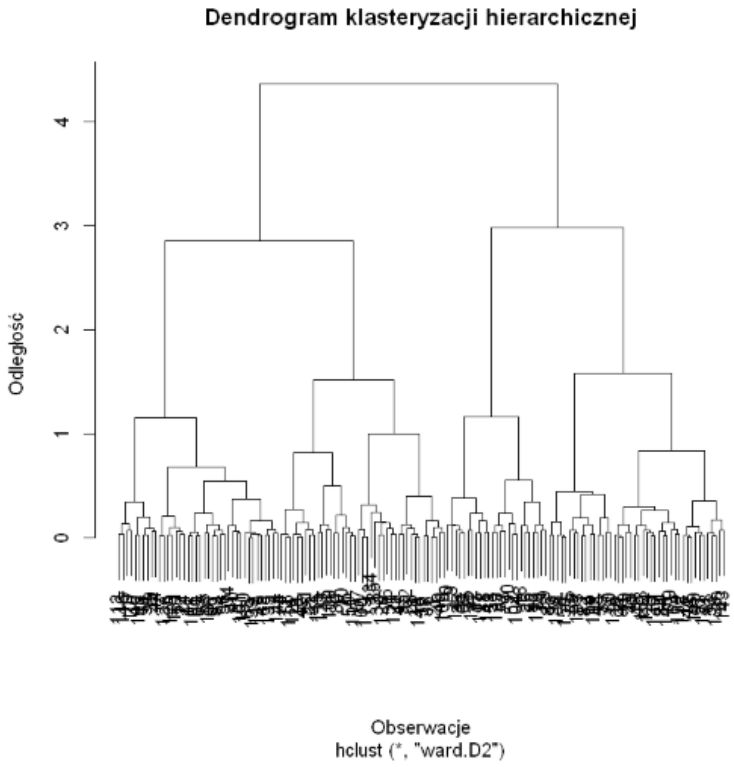


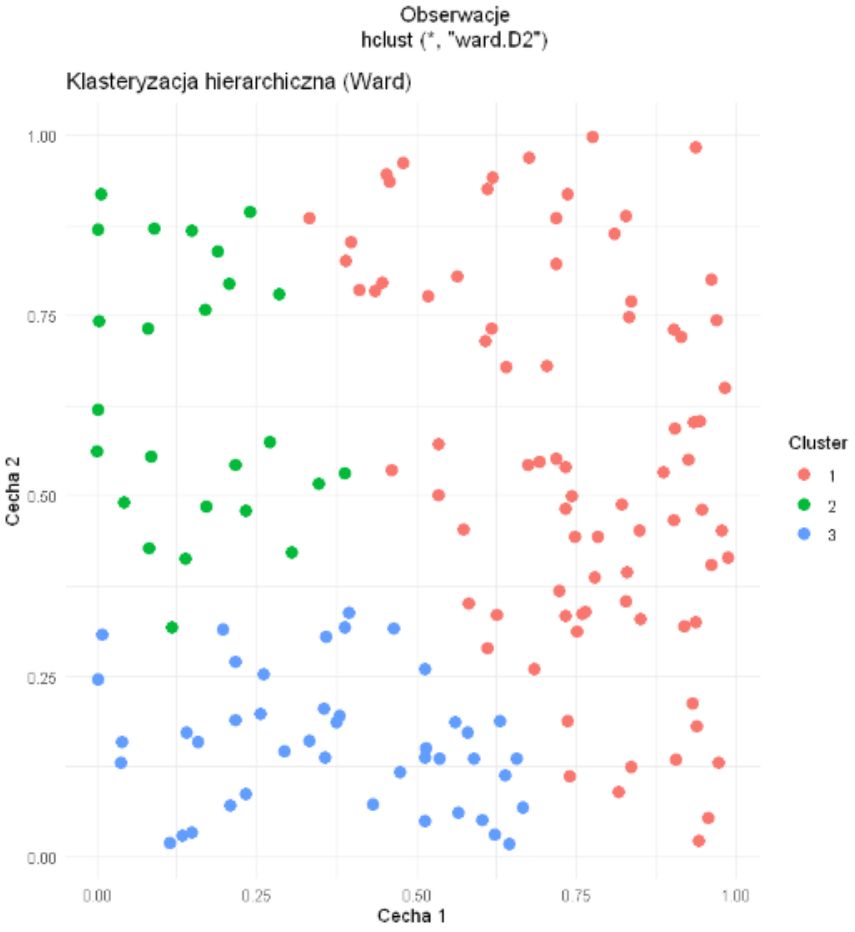


- R

```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 klastrów
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
Odległość
0
1
2
3
4
Obserwacje
hclust (*, "ward.D2")


Obserwacje
hclust (*, "ward.D2")
Klasteryzacja hierarchiczna (Ward)
Cecha 2
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
Cecha 1
Cluster
1
2
3

4. Wnioski

Uczenie nienadzorowane jest jednym z kluczowych typów uczenia maszynowego, w którym odkrywa się ukryte wzorce w danych bez potrzeby posiadania etykiet (czyli zmiennych wynikowych). Analiza skupień (ang. clustering) to proces grupowania danych w k klastrów, gdzie punkty w ramach jednego klastra są bardziej podobne do siebie niż do punktów z innych klastrów. Celem klasteryzacji jest minimalizacja funkcji kosztu J(C), która mierzy wewnętrzną różnorodność w klastrach. Ocena wyników klasteryzacji w uczeniu nienadzorowanym jest bardziej skomplikowana niż w przypadku uczenia nadzorowanego, ponieważ brak jest etykiet referencyjnych. Analiza głównych składowych (ang. Principal Component Analysis, PCA) to metoda redukcji wymiarowości danych, mająca na celu przekształcenie danych wielowymiarowych w mniejszą liczbę wymiarów przy zachowaniu maksymalnej ilości informacji. Metody niehierarchiczne do analizy skupień, takie jak k-means, polegają na podziale danych na k skupień, gdzie k jest parametrem wybranym przez użytkownika. Metody hierarchiczne budują hierarchię klastrów, umożliwiając zrozumienie struktury danych na różnych poziomach szczegółowości. Metryka odległości (np. odległość euklidesowa, Manhattan, kosinusowa) określa sposób mierzenia odległości między punktami, natomiast metoda łączenia (np. single linkage, complete linkage, average linkage) określa, jak mierzyć odległość między klastrami.