Przetwarzanie i Analiza Danych

Laboratorium 8: Analiza głównych składowych PCA

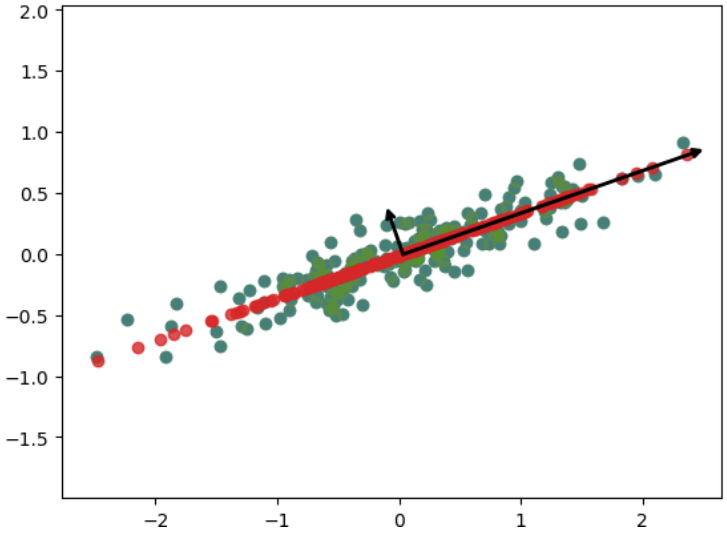
Cel ćwiczenia

Celem ćwiczenia jest implementacja algorytmu Analizy Głównych Składowych PCA dla danych wielowymiarowych. Należy wykonać własną implementację w formie funkcji `wiPCA`, która przyjmie na wejściu zbiór danych w formie macierzy oraz liczby wymiarów docelowych. Na wyjściu powinna znaleźć się macierz zerdukowanej przestrzeni cech, macierz wektorów własnych, wektor liczb własnych oraz dodatkowo średni wektor wejściowy. Napisana własnoręcznie funkcję porównać z implementacją dostępną w `sklearn`, np.

```
from sklearn import datasets
from sklearn.decomposition import PCA
iris = datasets.load_iris()
pca = PCA(n_components=2)
X_r = pca.fit(X).transform(X)
```

Zadania

1. Implementacja PCA i funkcji dwuwymiarowej wizualizacji przestrzeni cech
   (a) wygenerować w sposób losowy zbiór 200 obiektów dwuwymiarowych za pomocą funkcji z `numpy dot` i `rand` lub `randn`
   (b) zwizualizować obiekty na pomocą funkcji `matplotlib`, np. `scatter`
   (c) dokonać redukcji do jednego wymiaru za pomocą własnej funkcji `wiPCA` i zwizualizować wektory własne oraz rzut wygenerowanych obiektów na pierwszą składową, w sposób podobny do Rysunku 1
2. Testowanie PCA na zbiorze `iris`
   (a) Wczytać zbiór `iris` (sposób j.w.)



Rysunek 1: Przykładowa wizualizacja przestrzeni cech i wektorów własnych

(b) dokonać redukcji wymiarowości wszystkich obiektów w zbiorze do 2 najbardziej znaczących wymiarów za pomocą opracowanej funkcji `wiPCA`

(c) Zwizualizować elementy zbioru w przestrzeni cech z oznaczonymi klasami, np. za pomcą kolorów, etykiet lub symboli (`*, x, +, .`)

3. Testowanie PCA na zbiorze `digits`

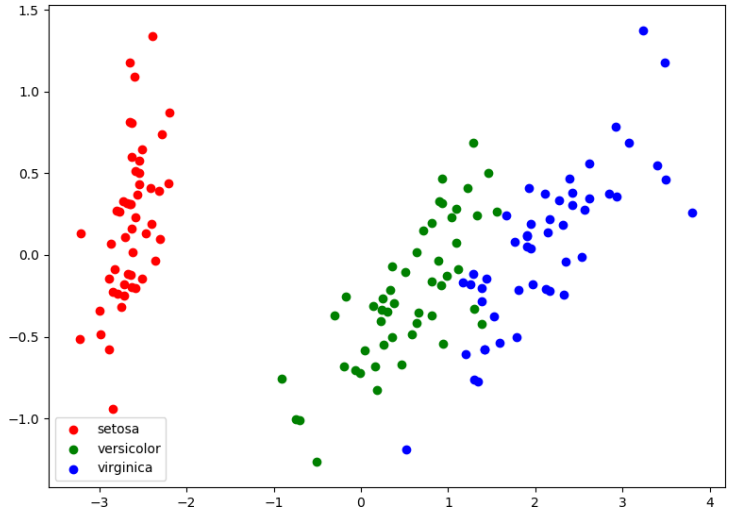
(a) Wczytać zbiór `digits` (`load_digits`)

(b) dokonać redukcji wymiarowości wszystkich obiektów w zbiorze do 2 najbardziej znaczących wymiarów za pomocą opracowanej funkcji `wiPCA`

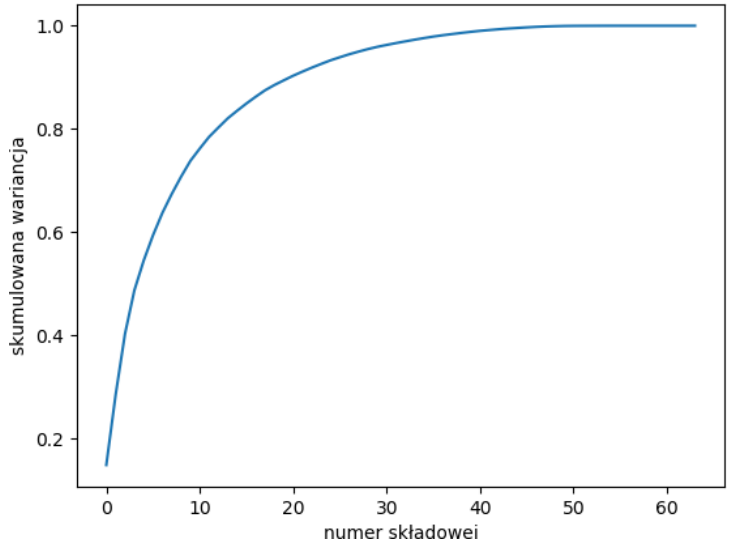
(c) Pokazać krzywą wariancji dla rosnącej liczby składowych głównych (tak jak na Rysunku 3)

(d) Zwizualizować elementy zbioru w przestrzeni cech z oznaczonymi klasami (podobnie do Rysunku 4, funkcja `scatter`)

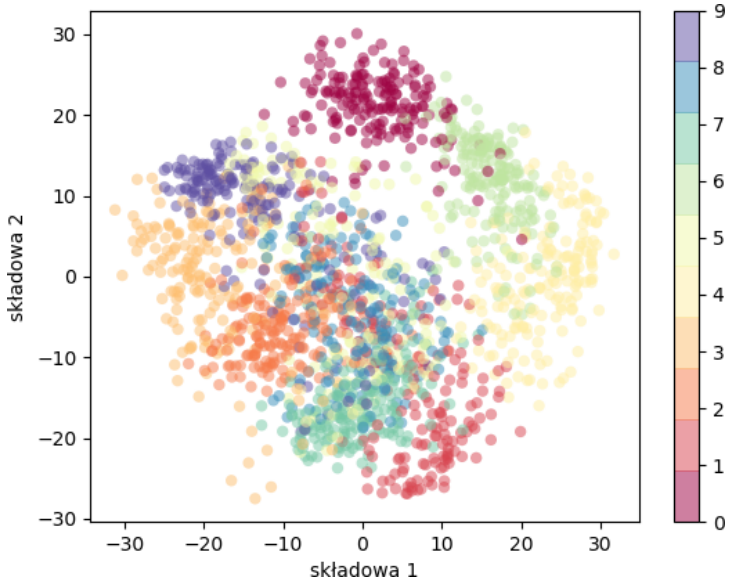
(e) Wykonać eksperyment polegający na ocenie średniego błędu rekonstrukcji dla całego zbioru dla kolejno zwiększającej się liczby składowych głównych (można to zrobić za pomoca obliczania odległości dla wszystkich obiektów w bazie od ich zrekonstruowanych postaci - funkcja z Laboratorium nr 6) - przykładowy przebieg zmianności odległości na Rysunku 5. Zadanie to wymaga napisania funkcji obliczającej transformatę odwrotną do PCA, zwracającą obiekt(-y) o wymiarowości zgodnej z obiektem(-ami)



Rysunek 2: Przykładowa wizualizacja obiektów z bazy `iris`

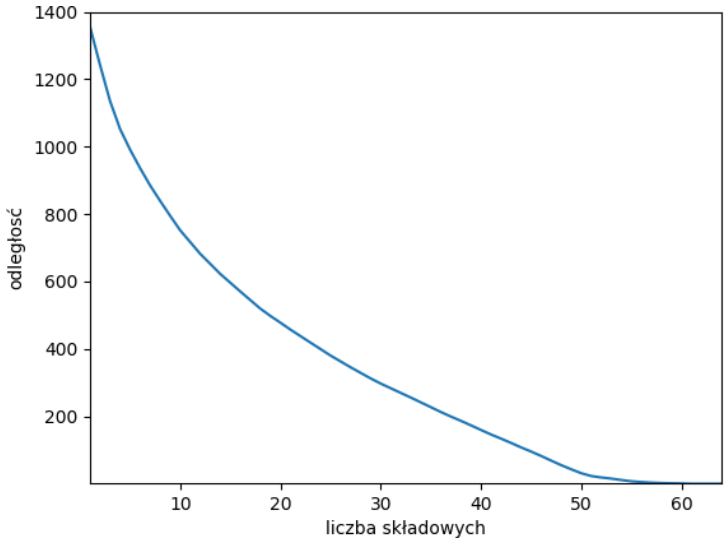


Rysunek 3: Przykładowa wizualizacja wariancji składowych głównych dla bazy `digits`



Rysunek 4: Przykładowa wizualizacja obiektów z bazy `digits`

wejściowymi (przed redukcją). *Dla ambitnych: pokazać zrekonstruowane cyfry dla 2, 4, 10 i 50 składowych głównych.*



Rysunek 5: Przykładowa wizualizacja róznicy pomiędzy obiektami oryginalnymi a zrekonstruowanymi