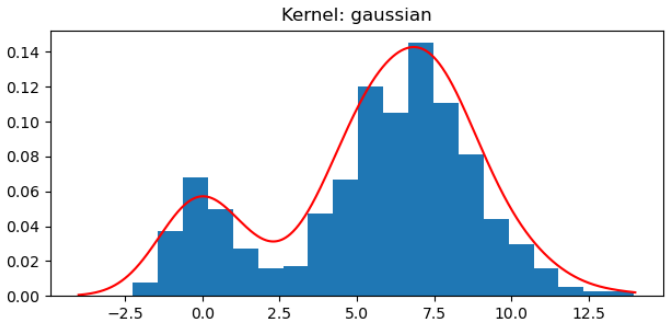
Laboratorium 4 — Wygładzanie danych, interpolacja, jądrowe estymatory funkcji gęstości.

Celem ćwiczenia jest zastosowanie jądrowych estymatorów funkcji gęstości, w celu wyznaczenia gęstości rozkładu dla wskazanych danych.

Zadanie 1 - jądrowy estymator gęstości (KDE)

Zaimportuj klasę `KernelDensity` z pakietu `sklearn.neighbors`, a następnie wykonaj następujące eksperymenty:

1. Za pomocą `np.random.randn` wygeneruj jednowymiarowy zbiór danych, zbiór ten powinien składać się z dwóch skupisk mniejszy zawierający 200 pomiarów zgodnych z rozkładem normalnym, oraz drugie większe składające się z 800 pomiarów wygenerowanych poprzez przesunięcie o 7 jednostek i przeskalowanie x2 próbek wygenerowanych z rozkładu normalnego;
2. Wyznacz histogram dla danych za pomocą polecenia `H, bins = np.histogram(X, bins = 20)`, wyznacz szerokość pojedynczego kosza. Przeskaluj wartości H tak aby suma wynosiła 1;
3. Utwórz obiekty KD, dla następujących funkcji: gaussian, tophat, linear, cosine, a następnie dopasuj modele do utworzonych danych;
4. Wygeneruj gęsty wektor (min 1000 elementów) z równomiernie rozłożonymi wartościami od min(X) do max(X), a następnie za pomocą metody `score_sample` wyznacz odpowiedzi modelu dla utworzonego wektora (pamiętaj, że utworzona metoda zwraca logarytm wiarygodność, w celu uzyskania prawdopodobieństwa zastosuje odpowiednie przekształcenie);
5. Dla każdego kernela narysuj histogram (wykorzystaj funkcję bar, pamiętaj o przesunięciu koszy o połowę ich szerokości, oraz ustawieniu szerokości słupków na szerokość kosza) wraz z uzyskaną funkcja gęstości, tak jak na rysunku 1. Która z funkcji dała najlepszy efekt dla tego zbioru danych?



Rysunek 1: Przykładowa wizualizacja działania algorytmu KDE.

Zadanie 2 - gussian mixture model (GMM)

Zaimportuj klasę `GaussianMixture` z pakietu `sklearn.mixture`, a następnie dla zbioru danych z zadania 1 wykonaj następujące eksperymenty:

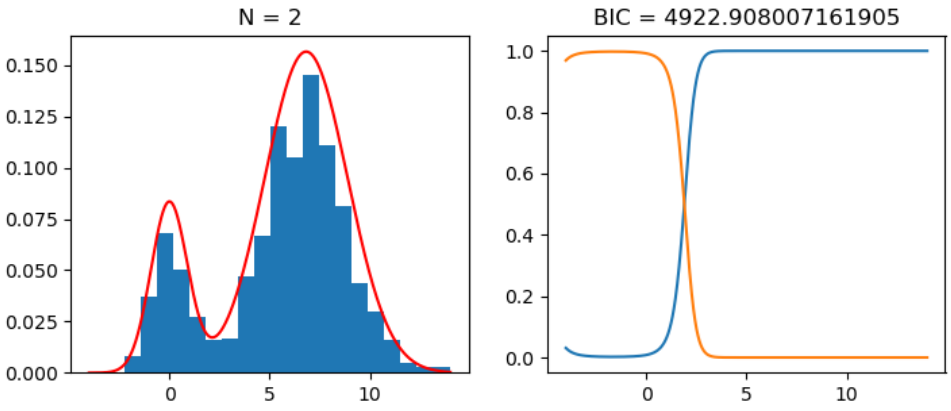
1. Utwórz obiekty GMM, dla liczby komponentów od 2 do 5 i tolerancji $1e-5$, a następnie dopasuj modele do utworzonych danych;

2. Za pomocą metody `score_sample` wyznacz wartości funkcji gęstości dla utworzonego wektora;

3. Za pomocą metody `predict_proba` wyznacz prawdopodobieństwo przynależności do poszczególnych klastrów.

4. Za pomocą odpowiedniej metody oblicz miarę BIC dla danych X.

5. Utwórz wykres zawierający histogram z funkcją gęstości, oraz obok wykres prawdopodobieństw, w tytule umieść liczbę komponentów oraz miarę BIC. Dla jakiej liczby komponentów uzyskano najlepsze dopasowanie? Czy miara BIC była w tym wypadku minimalna, czy maksymalna?



Rysunek 2: Przykładowa wizualizacja działania algorytmu GMM.

Zadanie 3 - gussian mixture model (GMM), klasteryzacja danych

Załaduj zbiór iris, a następnie:

1. Wykorzystaj algorytm PCA do redukcji wymiarowości zbioru iris, tak aby możliwe było zwizualizowanie działania algorytmu GMM;

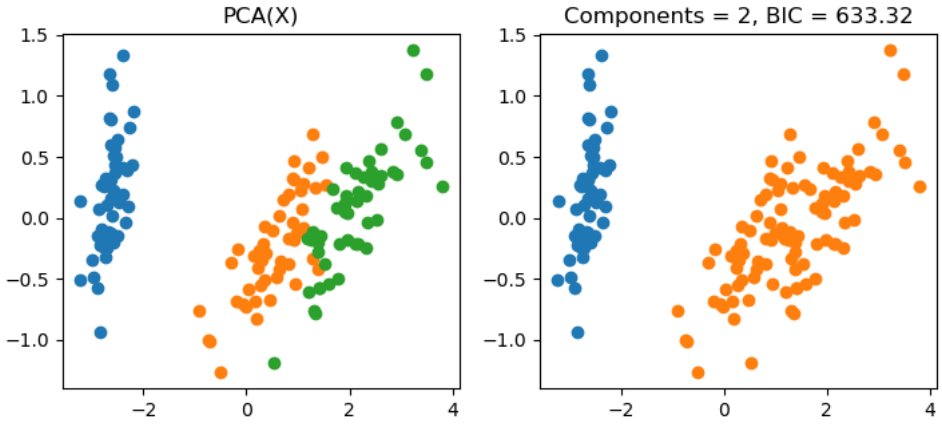
2. Utwórz obiekty GMM, dla liczby komponentów od 2 do 5 i tolerancji $1e-5$, a następnie dopasuj modele do zredukowanych danych;

3. Za pomocą odpowiedniej metody oblicz miarę BIC dla danych.

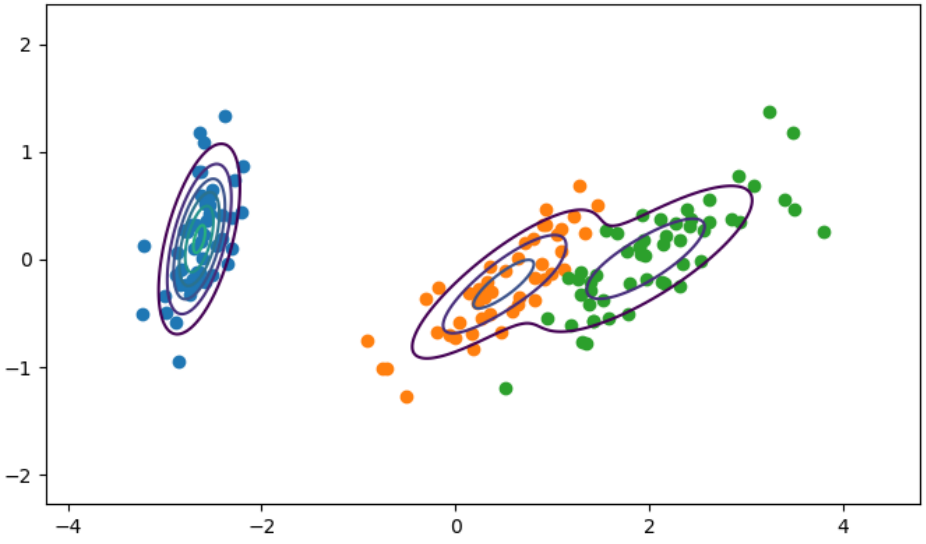
4. Przedstaw wyniki w jednym oknie, tak jak na rysunku 3, po lewej dane po transformacji PCA, gdzie różne kolory oznaczają klasę wczytaną z pliku, po prawej natomiast klasy powinny być zgodne z tymi uzyskanymi za pomocą metody `predict_proba` (pamiętaj, że metoda ta zwraca prawdopodobieństwo przynależności do każdej z klas, nas interesuje klas o największym prawdopodobieństwie dla danej próbki).

5. Czy w tym wypadku miara BIC także pozwoliła na wybór najlepszego modelu? Jeśli nie, co mogło być tego powodem?

6. Dla najlepiej dopasowanego modelu narysuj rozkład łączny dla klastrów (rysunek 4). W tym celu potrzebne będą takie funkcje jak `score_sample`, `contour` oraz `meshgrid`, poziomnice dla wykresu ustaw na: `np.arange(0.1, 1.0, 0.1)`. Pamiętaj, aby odpowiednio przekształcić wynik funkcji `score_sample`.



Rysunek 3: Przykładowa wizualizacja działania algorytmu GMM dla zbioru iris.



Rysunek 4: Rozkład łączny dla zbioru iris (algorytm GMM).