1. Wstęp

Celem ćwiczenia było zaimplementowanie od podstaw Naiwnego Klasyfikatora Bayesa przystosowanego do obsługi atrybutów ciągłych (Gaussian Naive Bayes) oraz porównanie jego skuteczności z gotową implementacją z biblioteki scikit-learn.

Jako główny problem testowy wykorzystano zbiór danych Seeds Data Set, zawierający pomiary geometryczne ziaren pszenicy należących do trzech różnych odmian: Kama, Rosa i Canadian.

Zbiór charakteryzuje się następującymi cechami:

- Liczba próbek: 210 (po 70 dla każdej z 3 klas)
- Liczba atrybutów: 7 wartości ciągłych (powierzchnia, obwód, współczynnik asymetrii)
- Brakujące dane: Brak (zbiór kompletny)

Dodatkowo, w celu weryfikacji poprawności algorytmu na prostszym problemie, przeprowadzono test na zbiorze Iris.

2. Opis Implementacji

Algorytm został zaimplementowany w języku Python. Kluczowe elementy rozwiązania to:

1. Reprezentacja wiedzy: Zamiast dyskretnych zliczeń, dla każdego atrybutu w każdej klasie obliczono średnią arytmetyczną oraz odchylenie standardowe. Pozwala to na przybliżenie rozkładu cech rozkładem normalnym (Gaussa).
2. Prawdopodobieństwo: Do obliczenia prawdopodobieństwa warunkowego wykorzystano funkcję gęstości prawdopodobieństwa rozkładu normalnego.
3. Stabilność numeryczna (Logarytmy): Zaimplementowano dwa warianty obliczania prawdopodobieństwa a posteriori:
    - Klasyczny: Mnożenie prawdopodobieństw (podatne na błędy zaokrągleń bliskich zera).
    - Logarytmiczny: Sumowanie logarytmów prawdopodobieństw. Jest to podejście zalecane w praktyce inżynierskiej, aby uniknąć problemu underflow (zaniku precyzji) przy dużej liczbie atrybutów.

3. Wyniki eksperymentów

3.1. Porównanie wariantu z logarytmowaniem i bez

Test przeprowadzono przy podziale zbioru 70% (trening) / 30% (test)

Wariant	Dokładność (Accuracy)	Komentarz
Bez logarytmowania	**88.89%**	Podstawowe mnożenie prawdopodobieństw
Z logarytmowaniem	**88.89%**	Sumowanie logarytmów

3.2. Wpływ proporcji podziału zbioru (Train/Test)

Proporcja (	Accuracy Trening	Accuracy Test	Wnioski
50 / 50	91.43%	87.62%	Model lekko niedouczony przez małą liczbę danych
60 / 40	92.06%	**90.48%**	**Najlepszy wynik. Optymalny balans danych**
70 / 30	92.52%	88.89%	Standardowy podział, stabilny wynik
80 / 20	93.45%	85.71%	Spadek na zbiorze testowym możliwe przetrenowanie lub specyficzny układ małego zbioru testowego

3.3. Porównanie z biblioteką Scikit-Learn

Implementacja	Accuracy	Uwagi
Własna	**88.89%**	Prosta implementacja z minimalnym wygładzaniem
Scikit-Learn	87.30%	Różnica wynika z domyślnego parametru var_smoothing w bibliotece

3.4. Test weryfikacyjny (Iris)

Dla zbioru Iris (klasyfikacja kwiatów) model osiągnął skuteczność **97.78%** na zbiorze testowym. Potwierdza to, że algorytm radzi sobie doskonale z dobrze separowalnymi danymi ciągłymi

4. Analiza wyników i wnioski

1. Skuteczność założenia Gaussa: Wysoka dokładność (89-90%) na zbiorze Seeds sugeruje, że parametry geometryczne ziaren mają rozkłady zbliżone do normalnego. Założenie naiwności (niezależności cech) nie przeszkodziło w uzyskaniu dobrego wyniku, mimo że parametry geometryczne (np. obwód i pole powierzchni) są ze sobą skorelowane.

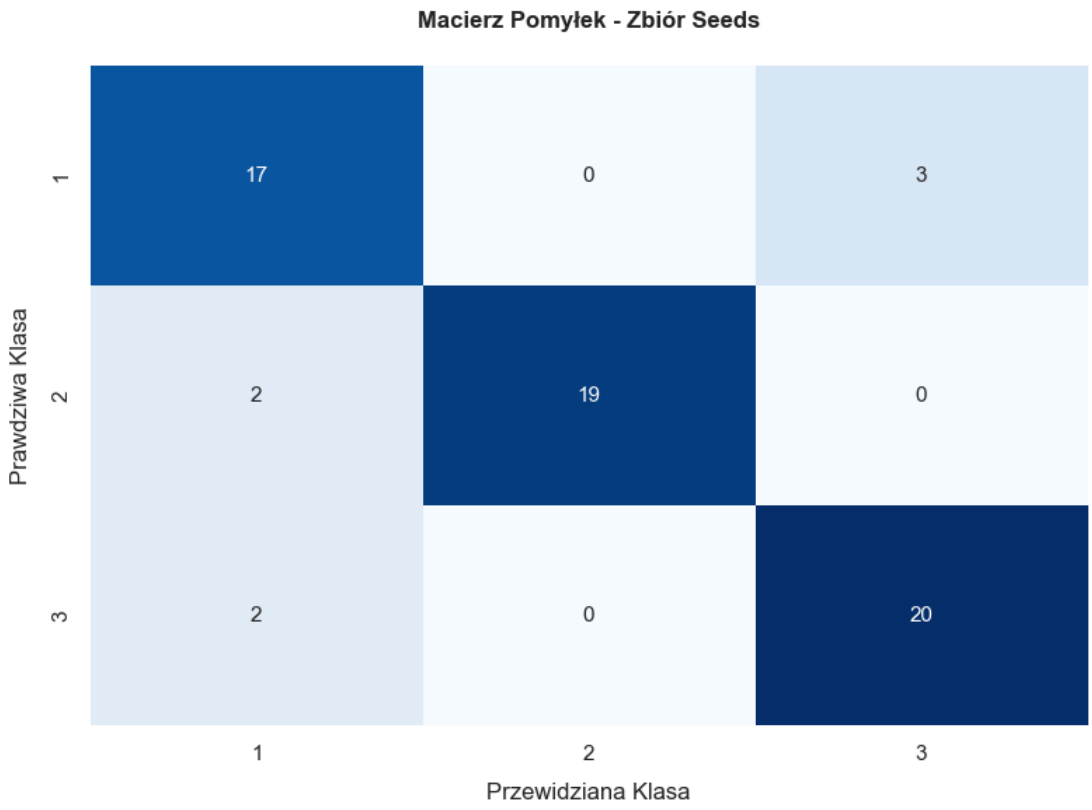
2. Przewaga własnej implementacji: W tym konkretnym przypadku "czysta" implementacja matematyczna okazała się minimalnie skuteczniejsza od bibliotecznej. Wynika to z faktu, że sklearn stosuje bardziej agresywne wygładzanie wariancji, co przy tak małym zbiorze danych mogło negatywnie wpłynąć na granice decyzyjne dla kilku granicznych próbek.

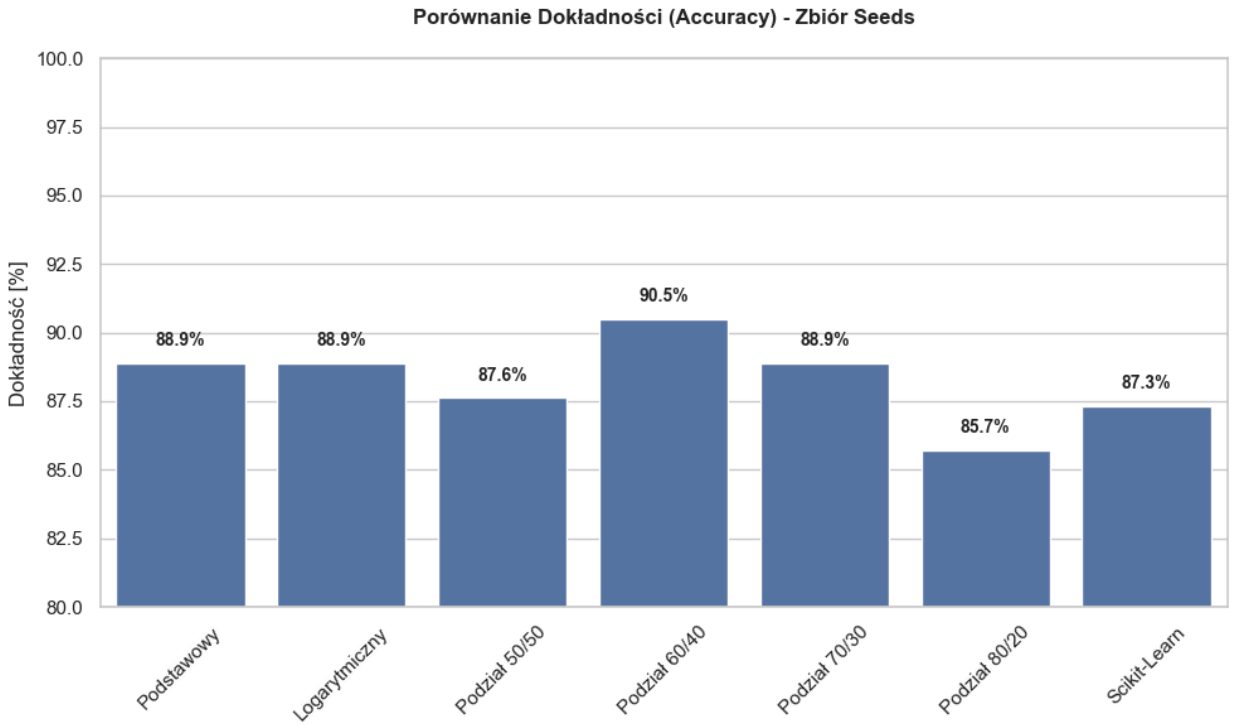
3. Czułość na podział danych: Zauważono, że przy podziale 80/20 wynik na zbiorze testowym spadł do 85.7%. Przy małych zbiorach danych (210 próbek) losowy dobór zbioru testowego ma duży wpływ na wynik końcowy. Najlepszą generalizację uzyskano przy podziale 60/40.

4. Analiza klas: Analizując macierz pomyłek (szczegółowe dane w acc_class), model najgorzej radził sobie z rozróżnianiem klasy 1 i 2 w niektórych podziałach, co sugeruje, że te odmiany pszenicy są do siebie najbardziej podobne pod względem wymiarów.

5. Podsumowanie

Zaimplementowany klasyfikator Gaussian Naive Bayes działa poprawnie i osiąga wyniki porównywalne lub lepsze od rozwiązań bibliotecznych na testowanych zbiorach. Zastosowanie logarytmów zabezpiecza algorytm numerycznie, a prostota modelu (brak hiperparametrów do strojenia) czyni go świetnym punktem odniesienia (baseline) dla bardziej złożonych metod.


Macierz Pomyłek - Zbiór Seeds
Prawdziwa Klasa
1
2
3
17
0
3
2
19
0
2
0
20
1
2
3
Przewidziana Klasa


Porównanie Dokładności (Accuracy) - Zbiór Seeds
Dokładność [%]
100.0
97.5
95.0
92.5
90.0
87.5
85.0
82.5
80.0
88.9%
88.9%
87.6%
90.5%
88.9%
85.7%
87.3%
Podstawowy
Logarytmiczny
Podział 50/50
Podział 60/40
Podział 70/30
Podział 80/20
Scikit-Learn