

# Sprawozdanie lab02

*Metody Inteligencji Obliczeniowej*

**Szymon Łabędziewski**

11.03.2025

## Spis treści

1.	Generowania danych dla klas K1 i K2 .....	2
2.	Analiza wpływu liczności zbioru uczącego na klasyfikację perceptronem.....	3
3.	Analiza klasyfikacji benzyny przy użyciu pojedynczego neuronu.....	4
4.	Klasyfikacja gatunków irysów za pomocą perceptronu .....	5
5.	Wpływ podziału danych na klasyfikację perceptronem.....	6
6.	Wpływ liczby epok na dokładność klasyfikacji.....	8

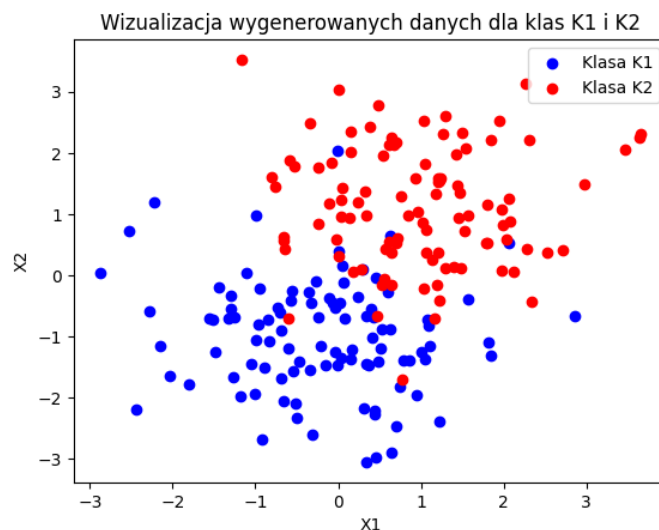
## 1. Generowania danych dla klas K1 i K2

Celem zadania było wygenerowanie zbioru danych składającego się z dwóch klas: K1 i K2. Punkty dla każdej klasy miały być losowane z rozkładów normalnych o określonych średnich i wariancji równej 1.

Klasyfikacja danych syntetycznych pozwala na testowanie algorytmów klasyfikacyjnych, takich jak perceptron. Odpowiednie rozmieszczenie punktów w przestrzeni cech ma kluczowe znaczenie dla poprawności późniejszej klasyfikacji.

Do wygenerowania danych wykorzystano funkcję *np.random.normal*, która pozwala na losowanie wartości z rozkładu normalnego.

- Klasa K1 została wygenerowana z rozkładu normalnego o średniej  $[0, -1]$  i wariancji 1.
- Klasa K2 została wygenerowana z rozkładu normalnego o średniej  $[1, 1]$  i wariancji 1.
- Wygenerowano po 100 punktów dla każdej klasy.
- Dane zostały zwizualizowane za pomocą *matplotlib.pyplot.scatter*, gdzie użyto różnych kolorów do oznaczenia każdej klasy.
- Następnie dane zostały połączone w jeden zbiór  $X$ , a etykiety klas zostały umieszczone w wektorze  $Y$  (0 dla K1, 1 dla K2).

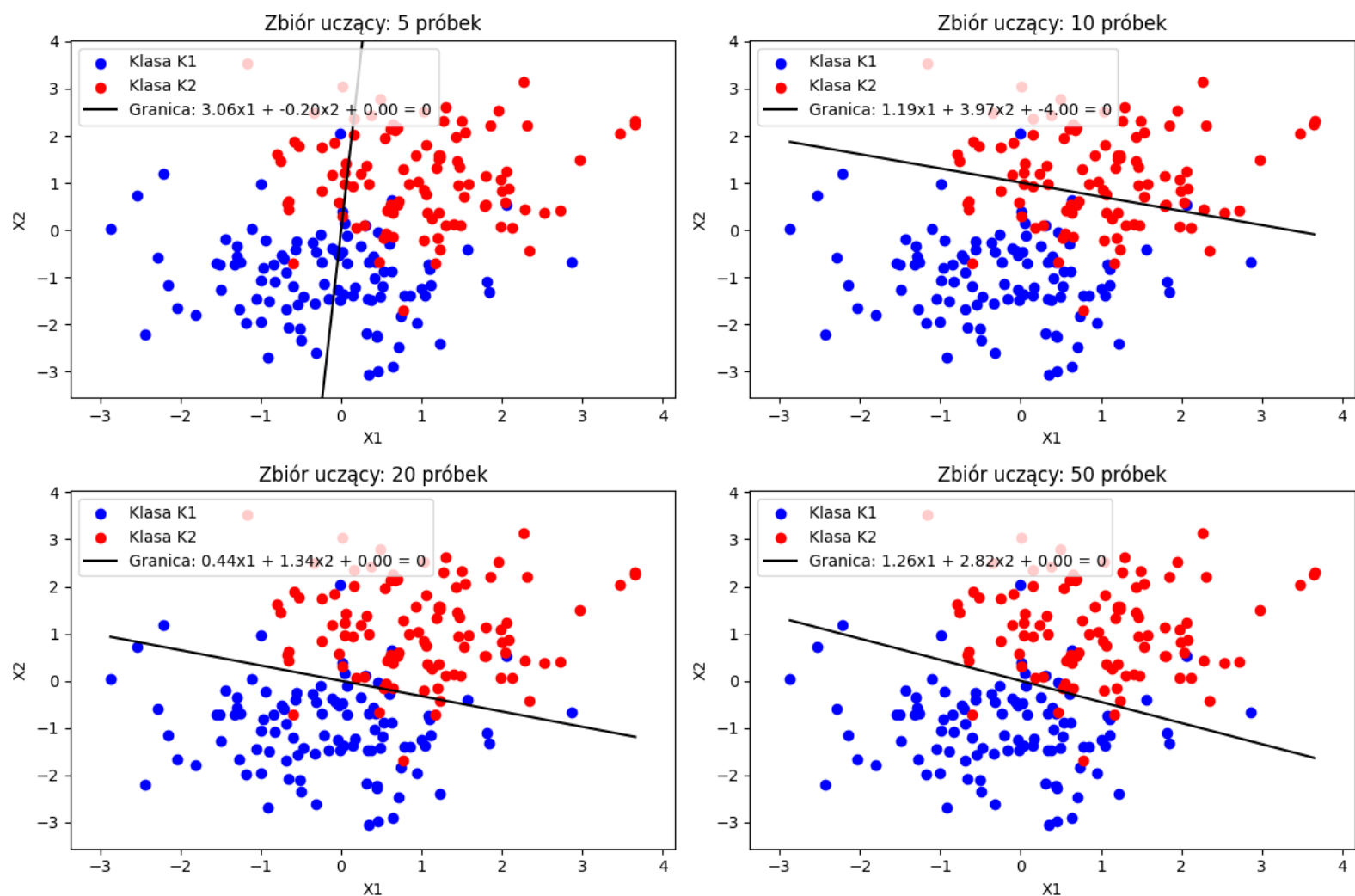


Wygenerowany wykres przedstawia dwie klasy, które są częściowo rozdzielne, ale z pewnym nakładaniem się. Klasyfikator będzie musiał nauczyć się odpowiedniej granicy decyzyjnej. Klasa K1 (niebieskie punkty) znajduje się głównie w dolnej części wykresu, natomiast klasa K2 (czerwone punkty) jest przesunięta w górę i w prawo, co odpowiada zadany parametrom rozkładu normalnego.

## 2. Analiza wpływu liczności zbioru uczącego na klasyfikację perceptronem

Celem tego etapu było zbadanie wpływu rozmiaru zbioru uczącego na jakość klasyfikacji za pomocą perceptronu. Przeprowadzono eksperyment dla różnych licznosci zbioru uczącego: 5, 10, 20 oraz 50 próbek, analizując uzyskane granice decyzyjne oraz skuteczność klasyfikacji.

- Dane podzielono na zbiór uczący oraz testowy w stosunku zależnym od wybranej liczby próbek uczących.
- Klasyfikator perceptronowy został wytrenowany dla każdego przypadku.
- Wyznaczono równania prostych oddzielających klasy na podstawie współczynników modelu.
- Skuteczność klasyfikatora została oceniona na zbiorze testowym.



- **Zbiór uczący: 5 próbek**  
Granica decyzyjna:  $3.06x_1 - 0.20x_2 + 0.00 = 0$   
Dokładność: **66%**  
Obserwacja: Granica decyzyjna jest niestabilna i może nie odzwierciedlać rzeczywistego podziału klas.
- **Zbiór uczący: 10 próbek**  
Granica decyzyjna:  $1.19x_1 + 3.97x_2 - 4.00 = 0$   
Dokładność: **81%**

Obserwacja: Granica zaczyna lepiej oddzielać klasy, choć nadal może być podatna na wpływ losowości próbek.

- **Zbiór uczący: 20 próbek**

Granica decyzyjna:  $0.44x_1 + 1.34x_2 + 0.00 = 0$

Dokładność: **92%**

Obserwacja: Widoczna poprawa w dopasowaniu granicy decyzyjnej do struktury danych.

- **Zbiór uczący: 50 próbek**

Granica decyzyjna:  $1.26x_1 + 2.82x_2 + 0.00 = 0$

Dokładność: **93%**

Obserwacja: Model osiąga wysoką skuteczność, a granica decyzyjna stabilizuje się, dobrze separując klasy.

Na wykresach przedstawiono punkty testowe oraz uzyskane granice decyzyjne dla każdego z przypadków. Widać, że dla małej liczby próbek (np. 5) granica jest nieregularna i niewystarczająco oddziela klasy, co skutkuje niższą dokładnością klasyfikacji. W miarę zwiększania liczby próbek granica staje się bardziej zgodna ze strukturą danych, co znajduje odzwierciedlenie w rosnącej dokładności modelu.

Wnioski:

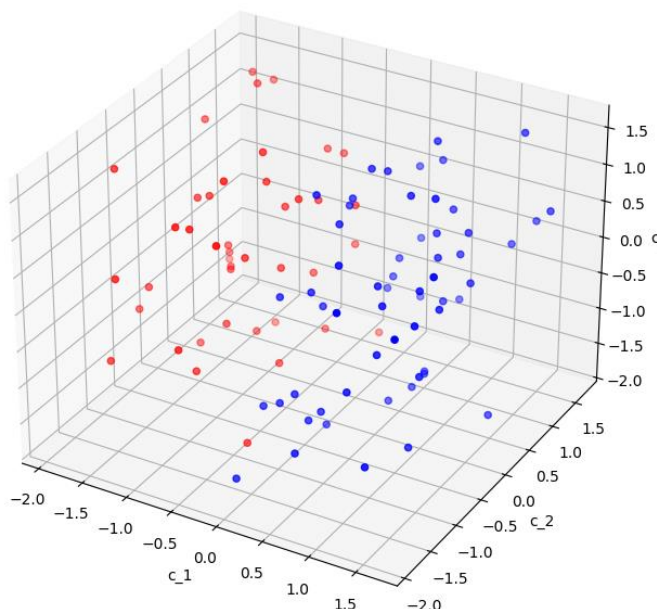
- Zbyt mały zbiór uczący prowadzi do słabej klasyfikacji i niestabilnych granic decyzyjnych.
- Wzrost liczby próbek pozwala na uzyskanie bardziej reprezentatywnej granicy decyzyjnej.
- Przy liczbie próbek 20 i więcej perceptron osiąga wysoką skuteczność klasyfikacji (>90%).
- Wizualizacja wyników potwierdza, że większa liczba próbek uczących prowadzi do stabilniejszej separacji klas.

### 3. Analiza klasyfikacji benzyny przy użyciu pojedynczego neuronu

Celem eksperymentu była analiza klasyfikacji próbek benzyny do dwóch klas czystości: A i B. W tym celu wykorzystano plik "fuel.txt", który zawierał trzy właściwości fizykochemiczne próbek oraz etykietę klasy. Klasy A i B zostały zakodowane numerycznie jako 1 i 0. Następnie przeprowadzono klasyfikację przy użyciu perceptronu, analizując skuteczność modelu w pięciokrotnym uczeniu sieci.

Dane zostały wczytane i poddane standaryzacji w celu zapewnienia równomiernej skali cech. W kolejnym etapie wykonano wizualizację przestrzenną danych przy użyciu wykresu 3D, co pozwoliło na ocenę ich rozkładu w przestrzeni trzech cech.

Scatter 3D - Fuel Data



Rys. 1 Wizualizacja przestrzenna danych z pliku projektu „fuel.txt”

Następnie przeprowadzono pięciokrotne uczenie perceptronu przy maksymalnej liczbie iteracji 100. W każdej epoce model był uczony na całym zbiorze danych, a następnie oceniana była jego skuteczność. Oprócz perceptronu wykorzystano również klasyfikator SVM z liniowym jądrem, który wyznaczył równanie granicy decyzyjnej.

**Dokładność w epoce 1: 0.97**  
Równanie granicy decyzyjnej:  $-3.0532 * c\_1 + 0.0317 * c\_2 + -0.0531 * c\_3 + (-0.6654) = 0$

**Dokładność w epoce 2: 0.97**  
Równanie granicy decyzyjnej:  $-3.0532 * c\_1 + 0.0317 * c\_2 + -0.0531 * c\_3 + (-0.6654) = 0$

**Dokładność w epoce 3: 0.97**  
Równanie granicy decyzyjnej:  $-3.0532 * c\_1 + 0.0317 * c\_2 + -0.0531 * c\_3 + (-0.6654) = 0$

**Dokładność w epoce 4: 0.97**  
Równanie granicy decyzyjnej:  $-3.0532 * c\_1 + 0.0317 * c\_2 + -0.0531 * c\_3 + (-0.6654) = 0$

**Dokładność w epoce 5: 0.97**  
Równanie granicy decyzyjnej:  $-3.0532 * c\_1 + 0.0317 * c\_2 + -0.0531 * c\_3 + (-0.6654) = 0$

Średnia dokładność po 5 próbach: 0.97

Każda z pięciu prób uczenia perceptronu wykazała wysoką dokładność klasyfikacji wynoszącą 97%. Równanie hiperpłaszczyzny decyzyjnej uzyskane przy użyciu modelu SVM pozostawało niezmiennie we wszystkich epokach.

Wysoka dokładność modelu wskazuje, że dane posiadają wyraźnie separowalne klasy, co pozwala na skuteczne rozdzielenie próbek przy pomocy liniowej granicy decyzyjnej. Ponieważ równanie płaszczyzny decyzyjnej pozostawało niezmiennie przez wszystkie epoki, można wnioskować, że perceptron osiągał stabilną zbieżność.

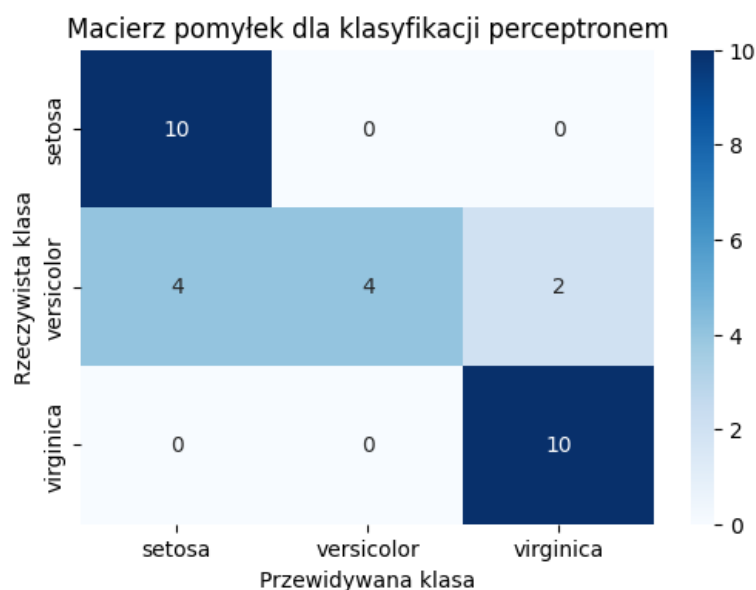
Wnioski:

- Perceptron skutecznie klasyfikuje próbki benzyny do odpowiednich klas z wysoką dokładnością.
- Uzyskana granica decyzyjna jest stabilna, co wskazuje na dobrze zdefiniowane separowalne klasy.

#### 4. Klasyfikacja gatunków irysów za pomocą perceptronu

Celem tego eksperymentu było zbudowanie klasyfikatora gatunków irysów na podstawie ich pomiarów przy użyciu perceptronu. Dane pochodziły ze zbioru *Iris*, który zawiera informacje o długości i szerokości działek kielicha oraz płatków dla trzech gatunków: *Setosa*, *Versicolor* i *Virginica*.

Zastosowano podział danych na zbiór uczący (80%) i zbiór testowy (20%), a następnie trenowano pojedynczą warstwę perceptronów przy użyciu funkcji `sklearn.linear_model.Perceptron`. Ostatecznym celem było sprawdzenie skuteczności modelu oraz analiza jego błędów poprzez macierz pomyłek.



Dokładność modelu wyniosła 80%, co oznacza, że perceptron poprawnie sklasyfikował 80% przypadków w zbiorze testowym.

Analiza macierzy pomyłek (przedstawionej na wykresie) wykazała:

- 100% poprawność klasyfikacji dla gatunku Setosa, co wskazuje na jego łatwą rozróżnialność.
- Większe trudności w klasyfikacji Versicolor i Virginica – model miał tendencję do błędnego przypisywania niektórych próbek klasy Versicolor do Setosa oraz do mylenia Versicolor z Virginica.

Wnioski

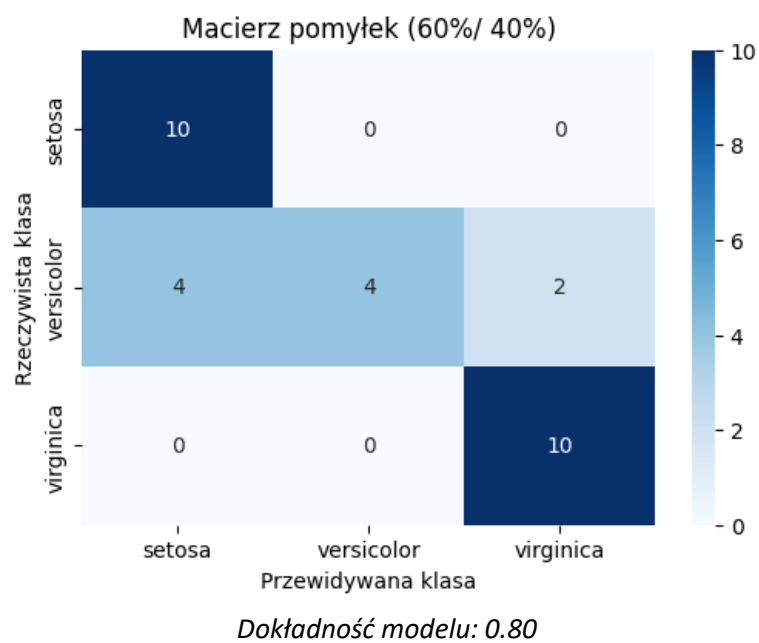
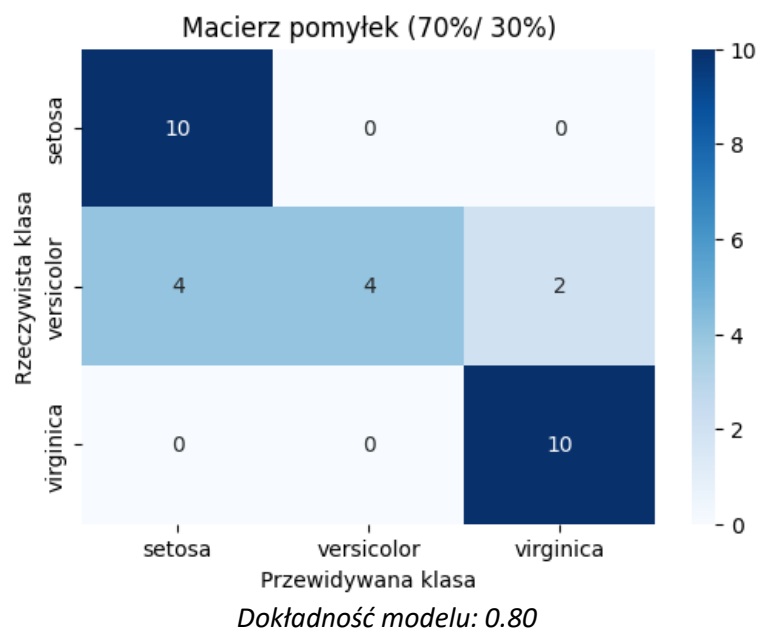
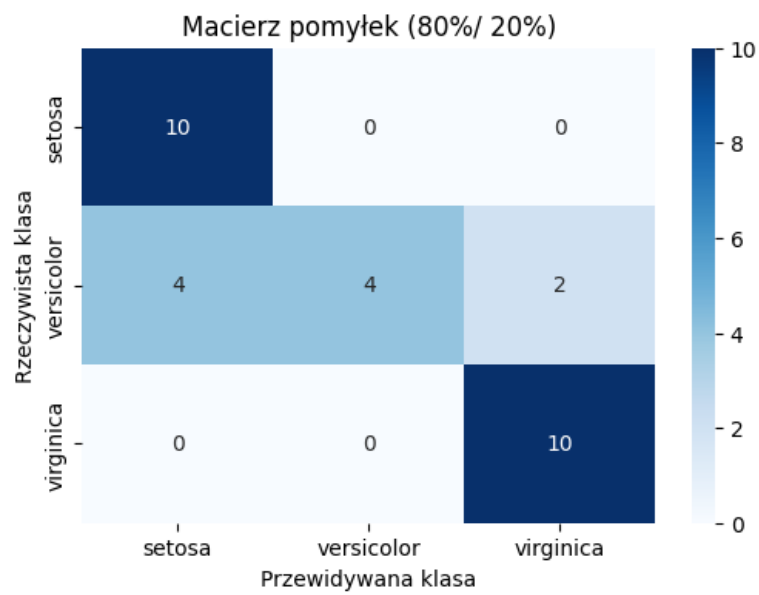
- Pojedyncza warstwa perceptronów dobrze radzi sobie z klasyfikacją Setosa, ale ma problemy z odróżnieniem Versicolor i Virginica, co obniża ogólną skuteczność modelu.
- Ograniczeniem perceptronu jest jego zdolność do klasyfikowania wyłącznie liniowo separowalnych zbiorów danych, co w przypadku zbioru Iris nie jest w pełni spełnione.

## 5. Wpływ podziału danych na klasyfikację perceptronem

Celem tego eksperymentu było sprawdzenie, jak różne podziały zbioru danych Iris na część uczącą i testową wpływają na skuteczność klasyfikacji za pomocą perceptronu. Przetestowano trzy różne podziały:

- 80% treningowe / 20% testowe,
- 70% treningowe / 30% testowe,
- 60% treningowe / 40% testowe.

Po każdym podziale trenowano model perceptronu i oceniano jego skuteczność poprzez dokładność klasyfikacji oraz analizę macierzy pomyłek.



Uzyskane dokładności dla wszystkich podziałów były identyczne i wyniosły 0.80.

Podobnie, macierze pomyłek dla wszystkich trzech przypadków miały dokładnie takie same wartości:

- Setosa została sklasyfikowana poprawnie w 100% przypadków.
- Versicolor była najtrudniejsza do klasyfikacji, często błędnie przypisywana jako Setosa lub Virginica.
- Virginica miała wysoką skuteczność klasyfikacji, ale pojawiały się pojedyncze błędne przypisania.

#### Wnioski

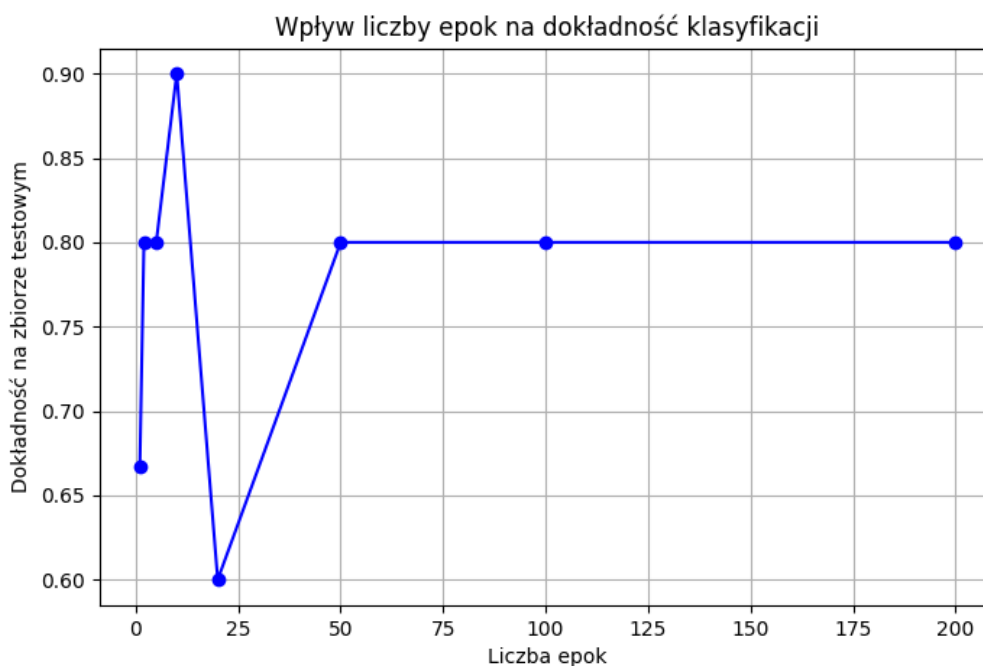
- Brak wpływu podziału danych na skuteczność klasyfikacji: wyniki sugerują, że perceptron osiąga tę samą dokładność niezależnie od proporcji podziału zbioru.
- Powtarzalność wyników może wynikać z tego, że perceptron wykorzystuje liniową separację, a głównym ograniczeniem modelu jest nieliniowość granicy decyzyjnej między Versicolor i Virginica.
- Potrzeba bardziej zaawansowanych modeli: bardziej skomplikowane podejścia mogłyby poprawić skuteczność klasyfikacji.

### 6. Wpływ liczby epok na dokładność klasyfikacji

Celem tego badania było sprawdzenie, jak liczba epok treningowych wpływa na dokładność klasyfikacji perceptronu dla zbioru danych Iris. Analizowano, ile faktycznie epok jest potrzebnych do osiągnięcia stabilnych wyników oraz czy zwiększanie liczby iteracji poprawia skuteczność klasyfikacji.

Epoki: 1	Dokładność: 0.67	Faktyczne epoki: 1
Epoki: 2	Dokładność: 0.80	Faktyczne epoki: 2
Epoki: 5	Dokładność: 0.80	Faktyczne epoki: 5
Epoki: 10	Dokładność: 0.90	Faktyczne epoki: 10
Epoki: 20	Dokładność: 0.60	Faktyczne epoki: 20
Epoki: 50	Dokładność: 0.80	Faktyczne epoki: 21
Epoki: 100	Dokładność: 0.80	Faktyczne epoki: 21
Epoki: 200	Dokładność: 0.80	Faktyczne epoki: 21





### Wyniki

- Dla 1 epoki model osiągnął dokładność 67%.
- Dla 2-5 epok model szybko ustabilizował się na poziomie 80%.
- Najwyższą dokładność (90%) uzyskano przy 10 epokach.
- Dalsze zwiększanie liczby epok nie poprawiło wyników, a dla 20 epok dokładność spadła do 60%.
- Dla 50 epok i więcej model stabilizował się na 80%, a liczba faktycznych epok przestała rosnąć (max. 21).

### Wnioski

- Wzrost liczby epok nie zawsze poprawia dokładność – model osiąga najlepsze wyniki w okolicach 10 epok.
- Zbyt duża liczba epok może pogorszyć klasyfikację.
- Perceptron osiąga zbieżność dość szybko – po 21 epokach liczba iteracji nie rośnie, co oznacza, że dalsze uczenie nie wnosi nowych informacji.
- Najbardziej efektywny zakres epok to 5-10, co zapewnia równowagę między dokładnością a czasem trenowania.