|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Piotr Poleszak | |  | 24/25 | PSI | | | | GC02 | |
|  | AUTOR | ROK AKAD. |  | | | SPEC. |  | GRUPA |
| Analiza zbiorów danych z zastosowaniem metod normalizacji i standaryzacji | | | | |  | | | 1 | |
| TEMAT | | | NR SPRAWOZDANIA | |
| 05.11.2024 | LMAD | | | | |  | | | |
| DATA WYKONANIA | PRZEDMIOT | | | | | PROWADZĄCY | | | |

# **Cel**

Celem niniejszego projektu jest przeprowadzenie analizy trzech różnych zbiorów danych z wykorzystaniem metod normalizacji i standaryzacji. Analiza obejmuje:

* **Zadanie 1**: Zbiór danych dotyczący trzech gatunków irysów.
* **Zadanie 2**: Widma Ramana zarejestrowane w czasie.
* **Zadanie 3**: Widma FTIR zarejestrowane w czasie.

Działania obejmują wczytanie i przygotowanie danych, zastosowanie odpowiednich metod normalizacji i standaryzacji oraz wizualizację i interpretację wyników.

# **Materiały i metody**

### **2.1. Zbiory danych**

* **Zbiór Iris**: Pobrany z [UCI Machine Learning Repository](https://archive.ics.uci.edu/dataset/53/iris). Zawiera 150 obserwacji trzech gatunków irysów (Iris-setosa, Iris-versicolor, Iris-virginica), opisanych za pomocą czterech cech: długości i szerokości kielicha oraz płatka (w centymetrach).
* **Widma Ramana**: Plik zawiera serie widm Ramana rejestrowanych w czasie. Pierwsza kolumna to liczba falowa cm-1, kolejne kolumny to intensywności dla poszczególnych pomiarów w czasie.
* **Widma FTIR**: Plik zawiera serie widm FTIR rejestrowanych w czasie. Pierwsza kolumna to liczba falowa cm-1, kolejne kolumny to absorbancja dla poszczególnych pomiarów.

### **2.2. Metody**

#### Wczytywanie i przygotowanie danych

* **Pandas**: Użyto biblioteki pandas do wczytywania i manipulacji danych.
* **Konwersja danych**: Zamieniono przecinki na kropki w wartościach numerycznych i przekonwertowano dane na typ float.

#### Analiza statystyczna (Zadanie 1)

* **Statystyki opisowe**: Użyto metody describe() do wygenerowania podstawowych statystyk dla cech.

#### Sprawdzenie kompletności danych (Zadanie 1)

* **Braki danych**: Użyto metody isnull().sum() do sprawdzenia brakujących wartości.

#### Normalizacja i standaryzacja

* **Normalizacja [0,1]**: Zastosowano MinMaxScaler(feature\_range=(0, 1)) z biblioteki scikit-learn.
* **Normalizacja [-1,1]**: Zastosowano MinMaxScaler(feature\_range=(-1, 1)).
* **Standaryzacja**: Zastosowano StandardScaler().

#### Normalizacja względem konkretnej wartości

* **Widma Ramana**: Każde widmo znormalizowano do amplitudy pasma przy 985 cm-1.
* **Widma FTIR**: Każde widmo znormalizowano do pola powierzchni pod wykresem (integracja metodą trapezów).

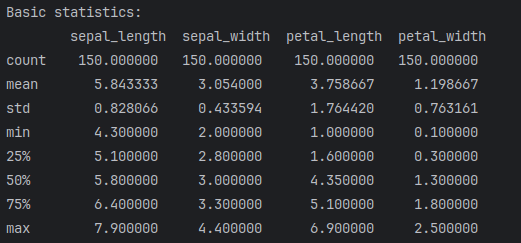
#### Wizualizacja

* **Matplotlib**: Użyto biblioteki matplotlib do tworzenia wykresów.
* **Wykresy zbiorcze**: Dla każdego zadania wygenerowano wykresy przed i po normalizacji.

# **Wyniki i dyskusja**

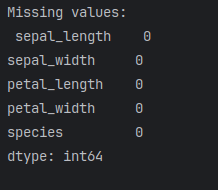
### **3.1. Zadanie 1: Analiza zbioru Iris**

#### **3.1.1. Statystyki podstawowe**

Podstawowe statystyki dla cech zbioru Iris:

#### **3.1.2. Sprawdzenie kompletności danych**

Brak brakujących wartości w danych:



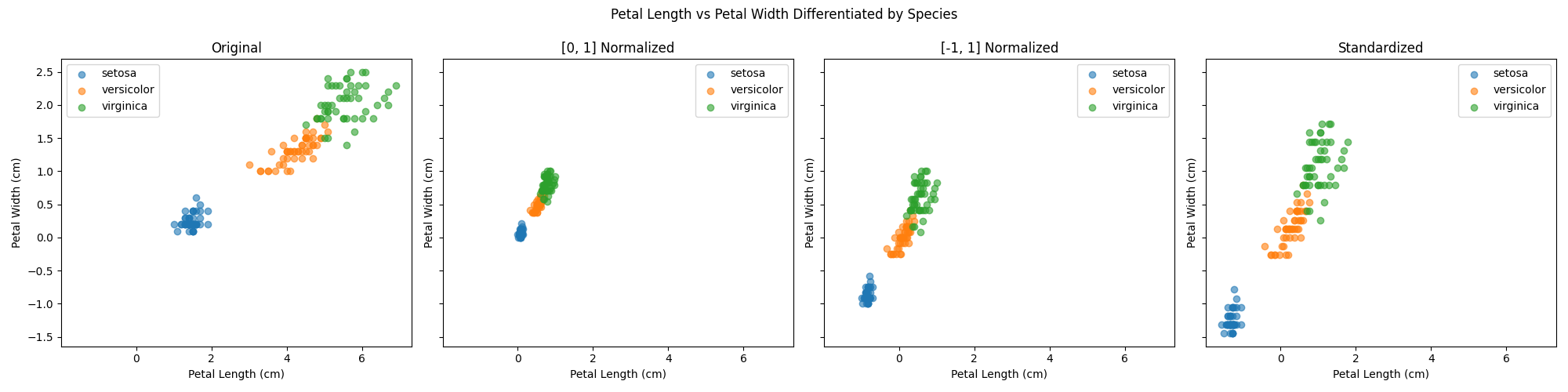
#### **3.1.3. Normalizacja i standaryzacja**

Dane zostały przekształcone za pomocą trzech metod:

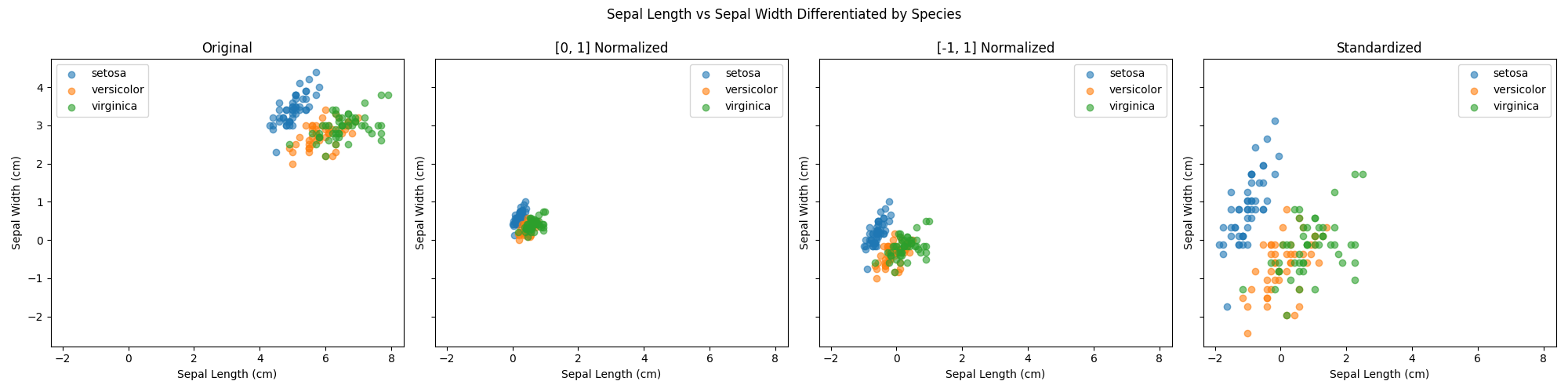
* **Normalizacja [0,1]**
* **Normalizacja [-1,1]**
* **Standaryzacja**

#### **3.1.4. Wizualizacja wyników**

**Zależność długości płatka od szerokości płatka zróżnicowana na podstawie gatunku:**



**Zależność długości kielicha od szerokości kielicha zróżnicowana na podstawie gatunku:**



**Dyskusja:**

Po normalizacji i standaryzacji dane zachowują swoją strukturę, ale skale osi ulegają zmianie. Dzięki temu można porównać cechy na ujednoliconej skali, co jest przydatne przy stosowaniu algorytmów uczących się, które są wrażliwe na skalę danych.

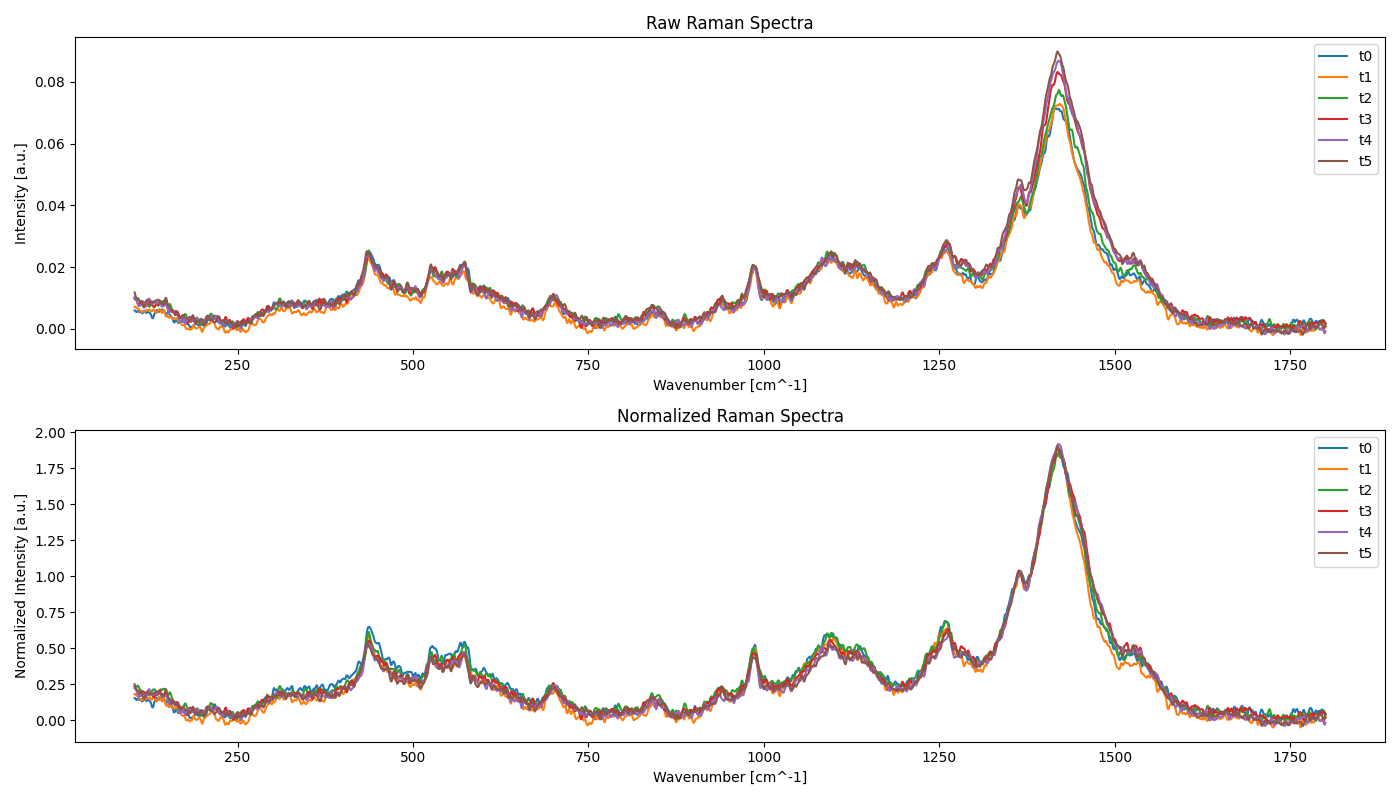
### **3.2. Zadanie 2: Widma Ramana**

#### **3.2.1. Normalizacja do amplitudy pasma przy 985 cm-1**

Wybrano pasmo przy 985 cm-1 jako punkt odniesienia do normalizacji, ponieważ jest ono charakterystyczne i obecne we wszystkich widmach. Każde widmo podzielono przez wartość intensywności przy tej liczbie falowej.

#### **3.2.2. Wizualizacja wyników**

**Widma przed normalizacją oraz po normalizacji:**



#### **3.2.3. Przyczyna zastosowania normalizacji**

Normalizacja względem konkretnego pasma kompensuje różnice w intensywności sygnału wynikające z warunków eksperymentalnych lub zmian w koncentracji próbki. Umożliwia to bezpośrednie porównanie kształtów widm i względnych intensywności innych pasm.

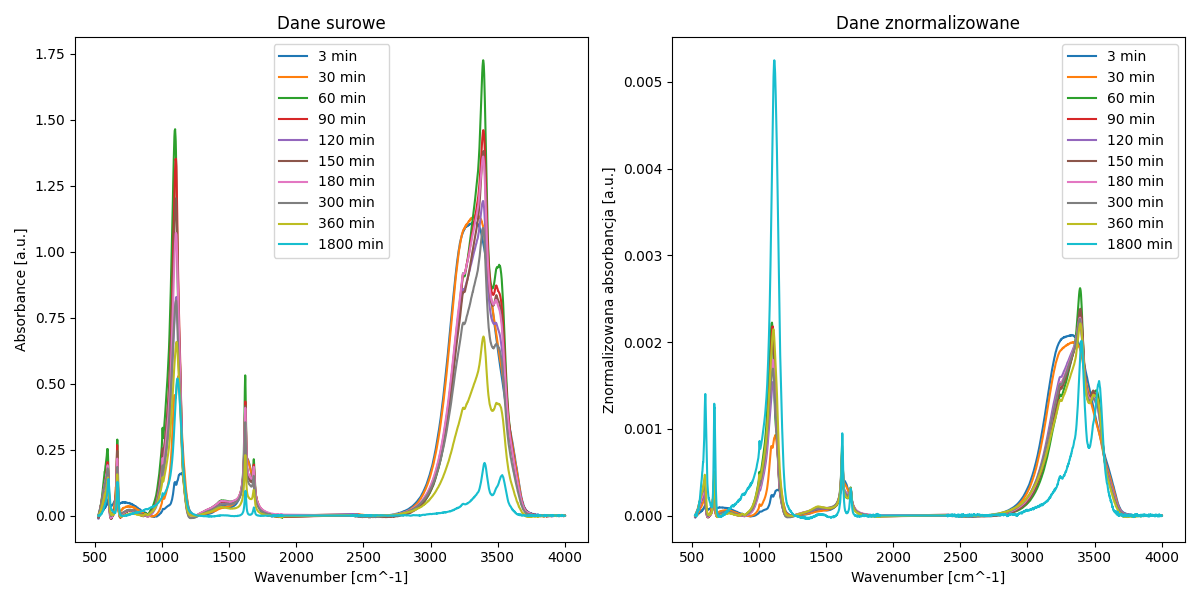
### **3.3. Zadanie 3: Widma FTIR**

#### **3.3.1. Normalizacja do pola powierzchni pod wykresem**

Każde widmo znormalizowano poprzez podzielenie przez całkowite pole powierzchni pod krzywą, obliczone metodą trapezów. Ta metoda uwzględnia całkowitą absorbancję próbki.

#### **3.3.2. Wizualizacja wyników**

**Widma przed normalizacją oraz po normalizacji:**



#### **3.3.3. Przyczyna zastosowania normalizacji**

Normalizacja do pola powierzchni eliminuje wpływ różnic w ogólnej intensywności sygnału, które mogą wynikać z różnych grubości próbek, nierówności powierzchni czy innych czynników eksperymentalnych. Pozwala to na porównanie względnych intensywności pasm między widmami.

# **Podsumowanie**

Przeprowadzona analiza wykazała, że zastosowanie odpowiednich metod normalizacji i standaryzacji jest ważne dla prawidłowej interpretacji danych. W przypadku zbioru Iris umożliwiło to porównanie cech o różnych skalach. Dla widm Ramana i FTIR normalizacja pozwoliła na kompensację różnic w intensywności sygnału do analizy jakościowej i ilościowej widm.

# **Bibliografia**

* Dua, D., Graff, C. (2019). UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science.
* Pedregosa, F., Varoquaux, G., Gramfort, A., i in. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
* Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90–95.
* McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 51–56.