# Algorytmy Uczenia Maszynowego

Sprawozdanie

| | | | |
|---|---|---|---|
| Data: | 28.03.2023 | Dzień: | Wtorek |
| Grupa: | grupa nr 2 | Godzina: | 18:55 |
| Nazwisko i imię: | Rydzyński Grzegorz | Prowadzący: | prof. dr hab. inż. Ewa Skubalska-Rafajłowicz |
| Nr indeksu: | 252958 | Temat: Sprawozdanie z programów | |

## Spis treści



## 1. Wstęp

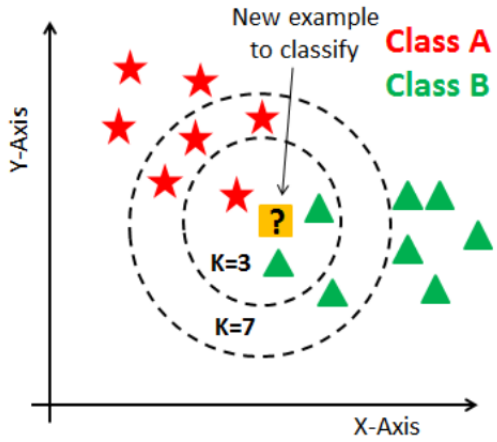
Na zajęciach projektowych należało zrealizować 3 programy realizujące zadanie klasyfikacji danych z wybranego zbioru. Należało zrealizować następujące algorytmy:

- algorytm k najbliższych sąsiadów,
- algorytm perceptronu Rosenblatta,
- wybrany algorytm klasyfikacji.

Pierwsze dwa algorytmy zostały zrealizowane na zbiorze danych Iris.

## 2. Algorytm k najbliższych sąsiadów

Algorytm k najbliższych sąsiadów jest jednym z nieparametrycznych algorytmów, wykorzystywanym do zadań regresji oraz klasyfikacji. Algorytm oblicza odległość danego punktu danych ze zbioru testowego do wszystkich pozostałych punktów danych ze zbioru treningowego. Po obliczeniu wszystkich odległości, wybierane jest k najbliższych punktów treningowych (nazywanych sąsiadami) do punktu testowego. Następnie, na podstawie ilości wystąpień danej klasy, do punktu testowego przypisywana jest klasa, która najczęściej wystąpiła wśród sąsiadów.




Rys. 1: Wizualizacja algorytmu odpowiednio dla k = 3 i k = 7

### 2.1. Działanie algorytmu - obliczanie odległości

Obliczanie odległości pomiędzy punktami wykonywane jest na podstawie różnych metryk długości. Najbardziej popularną metryką i metryką domyślnie wykorzystywaną przez biblioteki Scikit-Learn, (która została wykorzystana również w programie) jest metryka Euklidesowa, opisana wzorem:

$$d(p,q) = \sqrt{\sum (p_i - q_i)^2} \tag{1}$$

Do innych metryk można zaliczyć:

- metrykę Minkowskiego
- metrykę Czebyszewa
- metrykę "miejską" (metryka Manhattana)
- metrykę Minkowskiego

Z metryki Minkowskiego wynikają wszystkie powyższe metryki, ponieważ opisywana jest następująco:

$$d(p,q) = (\sum |p_i - q_i|^p)^{1/p} \tag{2}$$

Dla parametru p = 2 otrzymujemy metrykę Euklidesową, a dla p = 1 otrzymujemy metrykę Manhattana, którą można zapisać jako:

$$d(p,q) = \sum |p_i - q_i| \tag{3}$$

Dla parametru p dążącego do $\infty$ otrzymujemy metrykę Czebyszewa:

$$d(p,q) = \lim_{p\to\infty}(\sum |p_i - q_i|^p)^{1/p} = \max|p_i - q_i| \tag{4}$$

analogicznie dla p dążącego do $-\infty$ otrzymujemy:

$$d(p,q) = \lim_{p\to-\infty}(\sum |p_i - q_i|^p)^{1/p} = \min|p_i - q_i| \tag{5}$$

Implementacja funkcji wyliczającej dystans na podstawie metryki Euklidesowej wygląda następująco:

```
    @staticmethod
    def calculate_distance(test_row: list, train_row: list) -> float:
        """ This method calculates the euclidean distance between data points
        :param test_row: test_data point of type list
        :param train_row: train_data point of type list
        :return: euclidean distance between data points in the n-th dimension
        """
        euclidean_distance = 0.0
        for i in range(len(train_row)-1):
            euclidean_distance += (test_row[i] - train_row[i])**2
        return sqrt(euclidean_distance)
```

Listing 1: Implementacja funkcji wyliczającej dystans w programie

# 3. Testy

Dla algorytmu zostały przeprowadzone 3 serie testów, w celu sprawdzenia jakości predykcji.

## 3.1. Seria 1 - zmiana ilości sąsiadów dla stałego zbioru danych testowych

Pierwsza seria testów polega na zmianie ilości branych pod uwagę sąsiadów. Badany był zakres sąsiadów $n\_neighbors = (3, 8)$