Algorytm K-Najbliższych Sąsiadów (KNN)

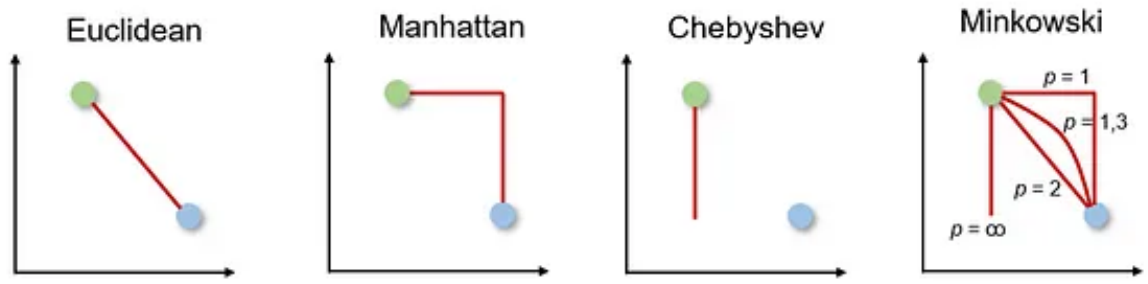
wstęp teoretyczny

1. *K-Nearest-Neighbors*

KNN (ang. *K-Nearest-Neighbors*) jest jednym z najprostszych algorytmów uczenia maszynowego, należącym do grupy algorytmów nadzorowanych. Jest również techniką nieparametryczną – w przeciwieństwie do metody MLR, która zakłada relację liniową pomiędzy zmiennymi objaśniającymi a modelowaną wielkością, nie ma ścisłych wymagań co do rozkładu i kształtu danych.
Znajduje zastosowanie zarówno w zadaniach klasyfikacji, jak i regresji, bazując na założeniu, że obiekty o podobnych cechach znajdują się blisko siebie w przestrzeni. Podstawowa idea KNN polega na przewidywaniu klasy lub wartości nowego punktu danych w oparciu o **k najbliższych sąsiadów** wybranych z zestawu treningowego.

Działanie algorytmu rozpoczyna się od określenia liczby sąsiadów k, a następnie obliczenia odległości między nowym punktem a wszystkimi punktami w zbiorze treningowym przy użyciu wybranej metryki, takiej jak odległość:

- Euklidesa
- Manhattan
- Czebyszewa
- Minkowskiego
- Canberra



Rysunek 1 https://towardsdatascience.com/9-distance-measures-in-data-science-918109d069fa

Odległość Euklidesowa (Euclidean distance):

Najbardziej powszechna miara odległości, definiująca rzeczywistą liniową odległość między dwoma punktami w przestrzeni n-wymiarowej.

$$d_{Euclidean}(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

gdzie p = $(p_1, p_2, \dots, p_n)$ *i* $q_1, q_2, \dots, q_n)$

Odległość Manhattan (Manhattan distance):

Odległość ta mierzy dystans jako sumę absolutnych różnic między współrzędnymi. Jest również znana jako odległość taksówkowa, gdyż przypomina dystans pokonywany na prostokątnej siatce ulic.

$$d_{Manhattan}(p,q) = \sum_{i=1}^{n} |p_i - q_i|$$

Odległość Czebyszewa (Chebyshev distance):

Odległość Czebyszewa to miara dystansu określająca maksymalną różnicę między odpowiadającymi sobie współrzędnymi dwóch wektorów. Jest często stosowana w geometrii dyskretnej.

$$d_{Chebyshev}(p,q) = \max_i |p_1 - q_1|$$

gdzie p = $(p_1, p_2, \dots, p_n)$ i $q_1, q_2, \dots, q_n)$ to dwa punkty w przestrzeni n-wymiarowej

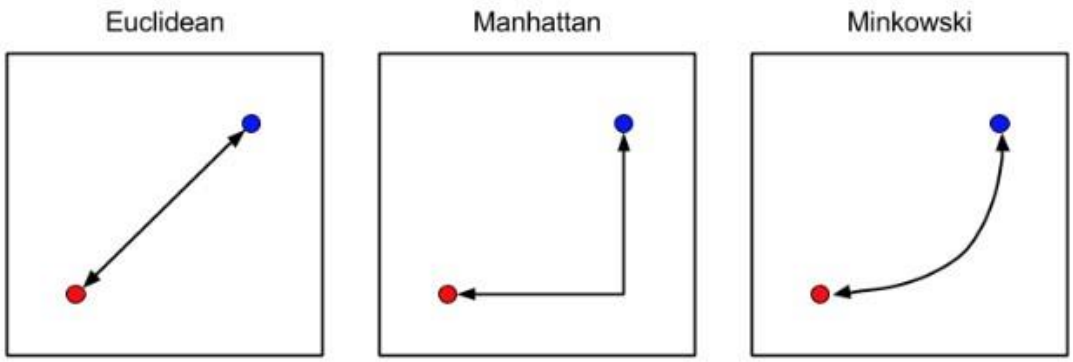
Odległość Minkowskiego (Minkowski distance)

Ogólny przypadek miar odległości, który zawiera w sobie zarówno Euklidesową, Manhattan, jak i inne miary w zależności od wartości parametru r.

$$d_{\text{Minkowski}}(p,q) = \left(\sum_{i=1}^{n} |p_i - q_i|^r \right)^{\frac{1}{r}}$$

Dla r =1: Odległość Manhattan,
Dla r =2: Odległość Euklidesowa,
Dla r → ∞: Odległość Chebyszewa

Parametr r pozwala dostosować miarę do różnych potrzeb



Odległość Canberra (Canberra distance):

Odległość Canberra to ważona miara odległości, w której różnice między współrzędnymi są dzielone przez sumę ich wartości absolutnych. Jest wrażliwa na wartości bliskie zeru i stosowana w analizie danych o dużym zakresie wartości.

$$d_{\text{Canberra}}(p,q) = \sum_{i=1}^{n} \frac{|p_i - q_i|}{|p_i| - |q_i|}$$

Przy założeniu, że $p_i + q_i \neq 0$. Jeśli $p_i + q_i = 0$, to odpowiedni jest pomijany.

2. Weighted KNN

Metoda KNN może zostać rozszerzona o ważenie wkładu sąsiadów, co prowadzi do tzw. **ważonego KNN (Weighted KNN)**. W tej wersji algorytmu sąsiedzi bliżej nowego punktu mają większy wpływ na przewidywanie niż sąsiedzi bardziej odlegli. Waga przypisywana każdemu sąsiadowi zależy zazwyczaj od jego odległości od punktu testowego. Najczęściej stosowanym podejściem jest przypisanie wag odwrotnie proporcjonalnych do odległości. Przykładowo, dla sąsiada *i*, waga w_i może być obliczana jako:

$$w_i = \frac{1}{d_i * \epsilon}$$

gdzie $\boldsymbol{d_i}$ to odległość między punktem testowym a sąsiadem $\boldsymbol{i}$, natomiast $\boldsymbol{\epsilon}$ to niewielka wartość zapobiegająca dzieleniu przez zero. Ma on znaczenie gdy mierzona jest odległość między tym samym punktem, a wówczas wynosi 0.

W przypadku **klasyfikacji**, każda klasa otrzymuje sumę wag swoich przedstawicieli wśród *k* najbliższych sąsiadów, a punkt testowy przypisywany jest do klasy o największej sumarycznej wadze.
Dla **regresji**, wartość punktu testowego jest wyznaczana jako ważona średnia wartości sąsiadów:

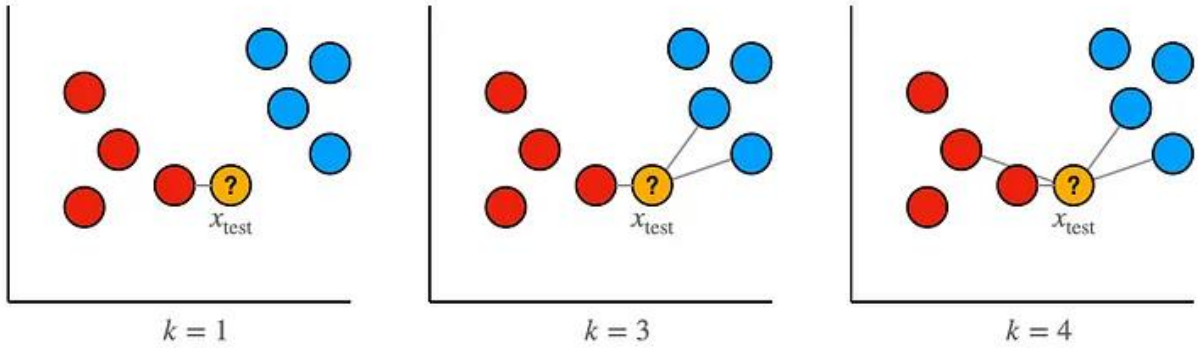
$$y_{pred} = \frac{\sum_{i=1}^{k} w_i \, y_i}{\sum_{i=1}^{k} w_i}$$

Gdzie y_i to wartość przypisana sąsiadowi *i,* a w_i to jego waga.

Ważony KNN jest szczególnie użyteczny w przypadkach, gdy dane są niejednorodne, a odległość od sąsiadów ma istotne znaczenie w kontekście modelowania. Cechuje się lepszą efektywnością w przypadku nierównomiernego rozmieszczenia danych, gdy bliżsi sąsiedzi są bardziej reprezentatywni. Zmniejsza wpływ odległych punktów, które mogą być nieistotne lub szumem w danych. Jednak skuteczność algorytmu zależy od odpowiedniego wyboru **miary odległości oraz funkcji wagowej.**

3. Wybór liczby *k*

Kolejnym krokiem jest wybranie *k* najbliższych punktów na podstawie obliczonych odległości. Na tej podstawie podejmowana jest decyzja: w przypadku klasyfikacji nowy punkt zostaje przypisany do klasy, która dominuje wśród jego sąsiadów, natomiast w przypadku regresji przewidywana wartość jest wyznaczana jako średnia lub mediana wartości sąsiadów. Liczba *k* wpływa na równowagę między **złożonością modelu** a jego zdolnością do generalizacji.



Rysunek 2 https://towardsdatascience.com/why-does-increasing-k-decrease-variance-in-knn-9ed6de2f5061

Wybór liczby *k* można dokonać w oparciu takie aspekty jak: wpływ liczby *k*, kryteria wyboru *k* oraz techniki wyboru *k*.

Wpływ liczby *k*:

Małe *k*:

- Model staje się bardziej szczegółowy, dopasowując się do lokalnych wzorców w danych.
- Ryzyko **przeuczenia** (overfitting), ponieważ decyzje mogą być zdominowane przez szum w danych.
- Dla $k=1$, algorytm klasyfikuje na podstawie najbliższego sąsiada, co może prowadzić do niestabilnych wyników.

Duże *k*:

- Model staje się bardziej uogólniony, uwzględniając większą liczbę punktów.
- Ryzyko **niedouczenia** (underfitting), gdy model staje się zbyt "gładki" i ignoruje istotne lokalne struktury danych.

Kryteria wyboru *k*:

Wielkość zbioru danych:

- Dla małych zbiorów danych mniejsze *k* mogą być odpowiednie, ale dla dużych zbiorów należy wybierać większe *k*, aby zminimalizować wpływ szumu.

Charakterystyka danych:

- Jeśli dane są gęsto rozmieszczone w przestrzeni, większe *k* lepiej uśrednią wpływ sąsiadów.
- Dla rzadkich lub rozproszonych danych mniejsze *k* mogą być bardziej odpowiednie.

Nieparzyste *k*:

- W przypadku klasyfikacji z wieloma klasami warto wybrać nieparzyste *k*, aby uniknąć sytuacji remisowych podczas głosowania większościowego.

Techniki doboru *k*:

Walidacja krzyżowa:

- Przetestowanie różnych wartości *k* na zbiorze walidacyjnym w celu znalezienia optymalnej liczby sąsiadów, która minimalizuje błąd klasyfikacji lub regresji.

Eksperymenty z danymi:

- Eksperymentalne testowanie różnych wartości k i wybór tej, która daje najlepsze wyniki na danych testowych.

Pro tip:

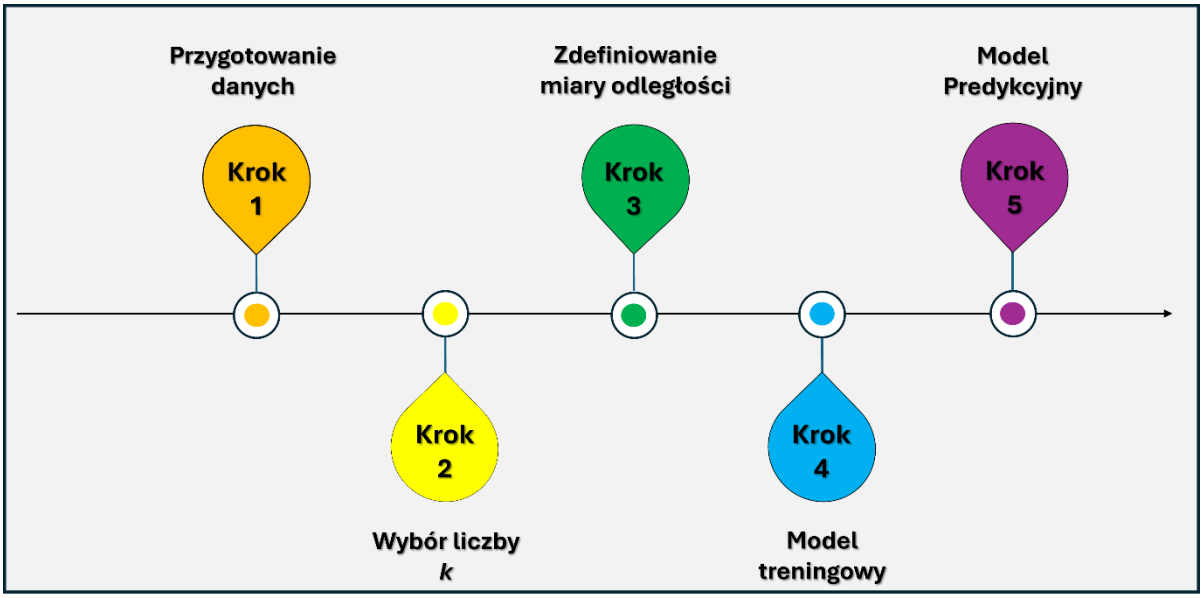
- Zacznij od $k = \sqrt{n}$, gdzie n to liczba próbek w zbiorze danych, i dostosuj wartość w zależności od wyników.
- Unikaj bardzo małych k, aby zmniejszyć wrażliwość modelu na szum.
- Testuj wartości k w szerokim zakresie, aby upewnić się, że wybrana liczba daje stabilne wyniki.

Metoda KNN ma wiele zalet, w tym prostotę implementacji i zdolność do radzenia sobie z danymi o nieliniowej strukturze. Niemniej jednak ma także swoje ograniczenia.

- Jest czasochłonna w przypadku dużych zbiorów danych, gdyż wymaga obliczenia odległości dla każdego punktu.
- Dodatkowo jej skuteczność zależy od odpowiedniego doboru parametru k oraz właściwej normalizacji danych, aby cechy o różnych skalach nie zaburzały wyników.

Mimo tych wyzwań KNN pozostaje popularnym algorytmem, szczególnie w sytuacjach, gdy dane są dobrze reprezentowane w wybranej przestrzeni cech, a zależności między punktami są łatwe do uchwycenia przez wybraną miarę odległości.

4. Algorytm KNN



1. **Zbiór danych: Przygotowanie i wstępna analiza**

   - Zbierz dane, które mają być użyte do treningu i testowania modelu.
   - Sprawdź jakość danych: uzupełnij brakujące wartości, usuń duplikaty i wyeliminuj szum, jeśli to możliwe.
   - Zidentyfikuj cechy (atrybuty) istotne dla analizy.

2. **Normalizacja danych**
   - Przeskaluj cechy do tego samego zakresu (np. 0–1 lub z-score), aby cechy o większych wartościach nie dominowały nad innymi podczas obliczania odległości.
   - Popularne metody skalowania: Min-Max, Standaryzacja (z-score).
3. **Wybór metryki odległości:**

   - Wybierz odpowiednią metrykę do obliczania odległości między punktami, np.:

     - Odległość Euklidesowa (gdy dane są ciągłe i mają naturalną interpretację geometryczną).
     - Odległość Manhattan (dla danych z dominującymi różnicami prostokątnymi).
     - Odległość Minkowskiego (umożliwia dostosowanie poprzez parametr rrr).

   - Upewnij się, że wybrana metryka odpowiada charakterystyce danych.

4. **Wybór liczby sąsiadów (*k*):**

   - Na początku ustaw *k* na niewielką wartość (np. k=3) i stopniowo ją dostosowuj.
   - Wykorzystaj walidację krzyżową, aby znaleźć optymalną wartość *k*, która minimalizuje błąd.
   - Rozważ nieparzyste *k*, aby uniknąć remisów w klasyfikacji.

5. **Podział danych na zbiory treningowe i testowe:**

   - Podziel dane na zbiór treningowy (np. 70–80% danych) oraz testowy (20–30%).
   - W razie potrzeby użyj walidacji krzyżowej, aby lepiej ocenić model.

6. **Implementacja algorytmu KNN:**

   - Dla każdego punktu testowego:
     a) Oblicz odległość między nim a każdym punktem w zbiorze treningowym.
     b) Posortuj punkty treningowe według rosnącej odległości.
     c) Wybierz *k* najbliższych sąsiadów.
     d) W przypadku klasyfikacji: określ klasę na podstawie głosowania większościowego.
     e) W przypadku regresji: oblicz średnią lub medianę wartości *k* sąsiadów.

7. **Ważenie sąsiadów (opcjonalne):**

   - Jeśli chcesz użyć ważonego KNN, przypisz wagę każdemu sąsiadowi na podstawie jego odległości od punktu testowego (np. im bliżej, tym większa waga).

8. **Ewaluacja modelu:**

   - Sprawdź dokładność modelu na zbiorze testowym, stosując odpowiednie miary, np.:

     a) Dla klasyfikacji: **macierz konfuzji**, dokładność, precyzja, czułość, F1-score.
     b) Dla regresji: **średni błąd kwadratowy (MSE)**, średni błąd absolutny (MAE).

   - Porównaj wyniki dla różnych wartości *k* i metryk odległości.

9. **Optymalizacja i dostrojenie modelu:**

   - Spróbuj różnych wartości *k*, metryk odległości lub metod ważenia, aby znaleźć najlepszą kombinację.
   - Przeanalizuj, czy dane wymagają dodatkowej obróbki (np. redukcji wymiarów, jeśli przestrzeń cech jest bardzo wysoka).

10. **Walidacja na nowych danych:**

    - Przetestuj model na nowych, niewidzianych wcześniej danych, aby ocenić jego zdolność do generalizacji.

5. Zadania:

Dane wejściowe – **ftalany.xlsx** zawiera dane dotyczące 32 ftalanów. Dane są już po autoskalowaniu. Podział na zbiór treningowy i walidacyjny znajdują się w poszczególnych arkuszach.

Zadanie 1:

a) Zbuduj model regresyjny *k*-najbliższych sąsiadów, którego zadaniem będzie przewidzenie stałej szybkości degradacji ftalanów w oparciu o podane deskryptory.

b) Dokonaj wyboru optymalnego *k*, oraz wykreśl wykres zależności *k* od RMSE, z uwzględnieniem „cross validation” cv = KFold(n_splits=?, shuffle=True, random_state=1). Wybierz optymalną wartość dla **n_splits** oraz uzasadnij swój wybór.

c) Przygotuj analizę KNN oraz oblicz statystyki R^2, RMSE, Q^2 i RMSEex. Narysuj wykres zależności y_{pred} od y_{obs}, oraz wykres słupkowy zależności y_{pred} od y_{obs}.

d) Oblicz statystyki R^2, RMSE, Q^2 i RMSEex dla modeli KNN z wykorzystaniem czterech dystansów: Euklidesa , Manhattan, Czebyszewa, Canberra, a następnie przygotuj wykres słupkowy ze statystykami dla każdej odległości. (Może to być jeden duży wykres).

Zadanie 2:
Powtórz podpunkty c) i d) dla KNN ważonego z uwzględnieniem czterech miar odległości.