



**Silesian  
University  
of Technology**

Dokumentacja projektowa

## **Systemy Sztucznej Inteligencji**

*Music Picker - Dobieranie najlepszych rekomendacji muzycznych*

Kierunek: InfPMS

Członkowie zespołu:

*Piotr Dusiński*

*Martyna Szandala*

Gliwice, 30 maja 2025

# Spis treści

<b>1</b>	<b>Wprowadzenie</b>	<b>2</b>
<b>2</b>	<b>Teoria do zrozumienia silnika</b>	<b>2</b>
2.1	Wprowadzenie do KNN . . . . .	2
2.2	Definicja . . . . .	2
2.3	Użyta metryka odległości w silniku . . . . .	3
<b>3</b>	<b>Silnik analizy danych</b>	<b>3</b>
3.1	Przygotowanie danych . . . . .	3
3.2	Model K-Najbliższych Sąsiadów . . . . .	4
3.3	Obliczanie średniego dystansu do sąsiadów . . . . .	4
3.4	Wyszukiwarka podobnych utworów . . . . .	5
<b>4</b>	<b>Moduł webowy</b>	<b>6</b>
<b>5</b>	<b>Przykład działania</b>	<b>6</b>
5.1	Formularz do wypełnienia . . . . .	6
5.2	Informacja zwrotna . . . . .	7
<b>6</b>	<b>Bibliografia</b>	<b>8</b>

# 1 Wprowadzenie

Niniejsza dokumentacja przedstawia system rekomendacji muzyki pod tytułem "Music Picker". Architektura projektu opiera się na częściach:

- Tworzenia silnika analizy danych
- Obsługi użytkownika w postaci modułu webowego

Razem te dwie cechy, które zostaną opisane w dalszej części dokumentacji, tworzą ten program.

## 2 Teoria do zrozumienia silnika

### 2.1 Wprowadzenie do KNN

Algorytm K-Najbliższych Sąsiadów (KNN) stanowi fundamentalną metodę w dziedzinie uczenia maszynowego i analizy danych, cenioną za swoją prostotę i efektywność. Jest to algorytm uczenia nadzorowanego, co oznacza, że do dokonywania przewidywań wykorzystuje dane oznaczone etykietami. Jego zastosowanie obejmuje zarówno zadania klasyfikacji, jak i regresji, co czyni go wszechstronnym narzędziem w wielu obszarach.

### 2.2 Definicja

KNN działa na zasadzie podobieństwa lub „bliskości”, przewidując etykietę lub wartość nowego punktu danych poprzez analizę etykiet lub wartości jego K-najbliższych sąsiadów w zbiorze treningowym. Podstawowe założenie algorytmu jest intuicyjne: podobne punkty danych znajdują się w bliskim sąsiedztwie w przestrzeni cech.

W przeciwieństwie do wielu innych algorytmów, które budują jawny model podczas fazy treningowej, KNN zapamiętuje cały zbiór danych treningowych i wykonuje obliczenia dopiero w momencie dokonywania przewidywań. Ta cecha pozwala mu szybko adaptować się do nowych danych bez konieczności ponownego trenowania całego modelu.

Ponadto, KNN jest algorytmem nieparametrycznym, co oznacza, że nie czyni żadnych założeń dotyczących bazowego rozkładu danych. Ta elastyczność umożliwia mu modelowanie złożonych, nieliniowych zależności w danych i sprawia, że jest on odpowiedni dla szerokiej gamy typów danych.

## 2.3 Użyta metryka odległości w silniku

W projekcie została użyta metoda Euklidesowa. Metrykę Euklidesową definiuje się jako odległość między  $x = (x - 1, \dots, x - n)$  a  $y = (y - 1, \dots, y - n)$ .

$$d(x, y) = \sqrt{(x - 1 - y - 1)^2 + \dots + (x - n - y - n)^2}$$

Gdy mamy już zdefiniowaną miarę podobieństwa oraz określoną liczbę grup  $k$ , dla każdego nowego obiektu  $y$  obliczamy jego odległość do każdego obiektu  $x$  znajdującego się w zbiorze treningowym. Spośród wszystkich tych obiektów  $x$  wybieramy  $k$  tych, które są najbliższe  $y$ . W ramach tej wybranej grupy  $k$  obiektów przeprowadzane jest głosowanie. Wartości (klasy lub etykiety), które pojawiają się najczęściej wśród tych  $k$  najbliższych sąsiadów, zostają przypisane do naszego obiektu  $y$ .

## 3 Silnik analizy danych

### 3.1 Przygotowanie danych

Na początku ze zbioru danych usuwane są wszystkie zduplikowane wiersze oraz rekordy zawierające brakujące wartości. Umożliwia to dalszą analizę na spójnym i kompletnym zbiorze danych. Następnie eliminowane są kolumny nieistotne z perspektywy analizy rekomendacji muzyki, takie jak: 'Unnamed', '0', 'track-id', 'mode', 'duration-ms', 'time-signature'. Pozostają wyłącznie te cechy, które są istotne dla algorytmów rekomendacyjnych. Ze zbioru danych usuwane są utwory o bardzo niskiej popularności, tj. te, których wartość w kolumnie 'popularity' mieści się w zakresie od 0 do 15. Dzięki temu model rekomendacyjny skupia się na utworach, które mają realną szansę zainteresować użytkowników. Wszystkie wybrane cechy numeryczne ('danceability', 'energy', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'loudness', 'tempo', 'popularity') są skalowane za pomocą standaryzacji (średnia = 0, odchylenie standardowe = 1) z wykorzystaniem klasy 'StandardScaler' z biblioteki 'sklearn'. Umożliwia to ich porównywanie oraz użycie w modelach takich jak K-Najbliższych Sąsiadów. Ostatecznie, oczyszczony i przeskalowany zbiór danych jest zapisywany do pliku 'data/spotify-data-cleaned.csv'. Dane te stanowią bazę dla kolejnych etapów analizy i rekomendacji w systemie. Powyższe kroki zapewniają, że dane wejściowe do systemu rekomendacyjnego są wysokiej jakości i gotowe do dalszego przetwarzania przez moduły analityczny.

### 3.2 Model K-Najbliższych Sąsiadów

System wykorzystuje model K-Najbliższych Sąsiadów (KNN) do analizy podobieństwa utworów muzycznych na podstawie ich cech numerycznych. Model ten został zaimplementowany w pliku `'recomender-system/KNN.py'`. Do uczenia modelu KNN wykorzystywany jest zestaw cech opisujących utwory muzyczne: `'danceability'`, `'energy'`, `'speechiness'`, `'acousticness'`, `'instrumentalness'`, `'liveness'`, `'valence'`, `'loudness'`, `'tempo'`. Te cechy zostały wybrane jako najistotniejsze dla określenia podobieństwa pomiędzy utworami. Dane wejściowe do modelu KNN są przygotowywane poprzez wybór powyższych cech dla każdego utworu oraz usunięcie wierszy z brakującymi wartościami. Proces ten zapewnia, że model pracuje na kompletnych i jednorodnie przygotowanych danych. Model KNN jest trenowany lub ładowany z pliku w zależności od dostępności gotowego modelu:

- Jeżeli istnieje zapisany na dysku model KNN z odpowiednią liczbą sąsiadów ( $k$ ), jest on wczytywany i wykorzystywany.
- Jeśli nie istnieje lub liczba sąsiadów nie zgadza się z oczekiwaniami, tworzony jest nowy model KNN, trenowany na aktualnych danych i zapisywany do pliku (`'data/knn-model-kk.pkl'`).

Liczba sąsiadów ( $k$ ) jest parametrem przekazywanym do funkcji ładującej/trenującej model. Pozwala to na łatwą konfigurację modelu w zależności od potrzeb analizy. Model KNN umożliwia efektywne wyszukiwanie utworów muzycznych podobnych do zadanego utworu na podstawie wybranych cech. Dzięki temu stanowi podstawę do generowania rekomendacji muzycznych w systemie. Implementacja modelu KNN opiera się na bibliotece `'scikit-learn'`, zapewniając wysoką wydajność oraz możliwość łatwego dostosowania parametrów modelu do wymagań użytkownika.

### 3.3 Obliczanie średniego dystansu do sąsiadów

W trakcie wyszukiwania podobnych utworów, system oblicza również średni dystans (ang. *average distance*) pomiędzy utworem wejściowym a zestawem proponowanych rekomendacji. Jest to miara opisująca, jak bardzo rekomendowane utwory są podobne do wybranego utworu pod względem cech numerycznych wykorzystywanych w modelu K-Najbliższych Sąsiadów (KNN). Po znalezieniu utworu wejściowego oraz pobraniu jego cech, model KNN wyszukuje zadeklarowaną liczbę najbliższych sąsiadów w przestrzeni cech (na podstawie wybranej metryki odległości, domyślnie euklidesowej). Funkcja `'kneighbors'` modelu KNN zwraca dwie wartości:

- tablicę dystansów ('distances') pomiędzy utworem wejściowym a każdym z najbliższych sąsiadów,
- indeksy odpowiadających im utworów w zbiorze danych.

Średni dystans wyliczany jest jako średnia arytmetyczna wszystkich wartości z tablicy dystansów dla wskazanej liczby rekomendacji (przed odfiltrowaniem utworu wejściowego, jeśli zostałby zwrócony przez model). Niższa wartość średniego dystansu oznacza, że rekomendowane utwory są bardzo podobne do utworu wejściowego. Wyższa wartość może sugerować, że w zbiorze nie ma utworów bardzo zbliżonych do wskazanego utworu lub utwór wejściowy jest nietypowy. Jeśli użytkownik poprosi o 10 rekomendacji, system wyznaczy 11 najbliższych sąsiadów (aby pominąć utwór wejściowy, jeśli się pojawi). Średni dystans zostanie obliczony na podstawie wartości zwróconych przez model KNN, zanim utwór wejściowy zostanie odfiltrowany z listy rekomendacji. Dzięki tej informacji użytkownik może łatwiej ocenić, na ile rekomendacje faktycznie odpowiadają jego oczekiwaniom pod względem podobieństwa muzycznego.

### 3.4 Wyszukiwarka podobnych utworów

Wyszukiwanie utworów muzycznych podobnych do zadanego utworu realizowane jest w module 'recomender-system/search-engine.py'. Moduł ten wykorzystuje wytrenowany model K-Najbliższych Sąsiadów (KNN) do odnajdywania rekomendacji na podstawie cech numerycznych utworów. Na początku w zbiorze danych wyszukiwany jest utwór odpowiadający podanym parametrom: tytułowi ('track-name') i wykonawcy ('artist-name'). Jeśli utwór nie zostanie znaleziony lub brakuje dla niego odpowiednich danych cechowych, zwracana jest pusta lista rekomendacji. Dla znalezionej utworu pobierane są jego cechy numeryczne, zgodnie z listą wykorzystywaną przez model KNN ('danceability', 'energy', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'loudness', 'tempo'). Cechy te muszą być kompletne. Wyszukiwanie podobnych utworów z użyciem KNN:

- Wyznaczana jest liczba sąsiadów ( $k$ ) równa liczbie oczekiwanych rekomendacji powiększonej o jeden (aby móc pominąć utwór wejściowy).
- Model KNN jest ładowany lub trenowany z odpowiednią wartością  $k$ .
- Dla cech wybranego utworu wyszukiwani są najbliżsi sąsiedzi (najbardziej podobne utwory) w przestrzeni cech.

Tworzenie listy rekomendacji:

- Z listy znalezionych sąsiadów usuwany jest utwór wejściowy, jeśli się w niej znajduje.
- Tworzona jest lista słowników zawierających tytuły i wykonawców rekomendowanych utworów.
- Liczba zwracanych rekomendacji odpowiada liczbie zadanej przez użytkownika.
- Funkcja zwraca także średni dystans pomiędzy utworem wejściowym a rekomendowanymi utworami, co pozwala oszacować „bliskość” rekomendacji.
- W przypadku błędu w trakcie wyszukiwania, zwracana jest informacja diagnostyczna.

Moduł ten stanowi kluczowy element systemu rekomendacji muzyki, umożliwiając użytkownikowi wyszukiwanie utworów najbardziej zbliżonych do podanego utworu na podstawie analizy cech muzycznych.

## 4 Moduł webowy

Do obsługi użytkownika biblioteki Flask - lekka platforma aplikacji internetowych WSGI. Forma prezentacyjna z formularzem zbiera informacje od klienta, gdzie następnie przekazuje informacje do silnika analizy danych. Gdy silnik skończy swoje procesy zwraca dane do backendu, który następnie wysyła informacje zwrotne w czytelny sposób do użytkownika.

## 5 Przykład działania

### 5.1 Formularz do wypełnienia

Użytkownik by dostać rekomendacje musi wypełnić puste pola, a następnie kliknąć "Szukaj"

Music Picker

Wpisz autora i utwór by zobaczyć spersonalizowane rekomendacje muzyczne!

Utwór

Artysta

Liczba podobnych utworów do wyświetlenia

Szukaj

## 5.2 Informacja zwrotna

Po odczekaniu przetwarzania procesu przez silnik, oraz po poprawnym wypełnieniu formularza pojawią się informacje zwrotne w postaci

Music Picker

Wpisz autora i utwór by zobaczyć spersonalizowane rekomendacje muzyczne!

Utwór

Chopin, Nocturne No. 1 in B-Flat Minor

Artysta

Frédéric Chopin

Liczba podobnych utworów do wyświetlenia

10

Szukaj

Średni dystans do potencjalnych rekomendacji: 0.1685

Rekomendowane utwory:

Tytuł utworu	Artysta
Love & Piano	Piano Peace
Nocturne No. 1 in B-Flat Minor, Op. 9 No. 1	Frédéric Chopin
Sonatine: II. Ballad	Friedrich Gulda
Saying Things	Emanuel Fremont
Straubish	Anne Straub



## 6 Bibliografia

### Literatura

- [1] <https://pandas.pydata.org>
- [2] <https://flask.palletsprojects.com/en/stable/>
- [3] <https://scikit-learn.org/stable/>
- [4] <https://numpy.org/>
- [5] <https://seaborn.pydata.org/>
- [6] <https://www.elastic.co/>
- [7] <https://neptune.ai/>
- [8] <https://www.wikipedia.org/>
- [9] <https://www.geeksforgeeks.org/>