AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
WYDZIAŁ ZARZĄDZANIA

Samodzielna Pracownia Zastosowań Matematyki w Ekonomii

Praca dyplomowa magisterska

Analiza porównawcza wybranych metod grupowania i klasyfikacji w systemach rekomendacji z wykorzystaniem uczenia maszynowego.

The comparison analysis of selected classification and grouping algorithms applied in recommendation systems using machine learning.

Autor: Roksana Świerczek
Kierunek studiów: Informatyka i ekonometria
Opiekun pracy: dr Tomasz Wójtowicz

Kraków, 2020

„Uprzedzony o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.): „Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystyczne wykonanie albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.”, a także uprzedzony o odpowiedzialności dyscyplinarnej na podstawie art. 211 ust. 1 ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (t.j. Dz. U. z 2012 r. poz. 572, z późn. zm.) „Za naruszenie przepisów obowiązujących w uczelni oraz za czyny uchybiające godności studenta student ponosi odpowiedzialność dyscyplinarną przed komisją dyscyplinarną albo przed sądem koleżeńskim samorządu studenckiego, zwanym dalej "sądem koleżeńskim"”, oświadczam, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i że nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.”

Spis treści

Wstęp

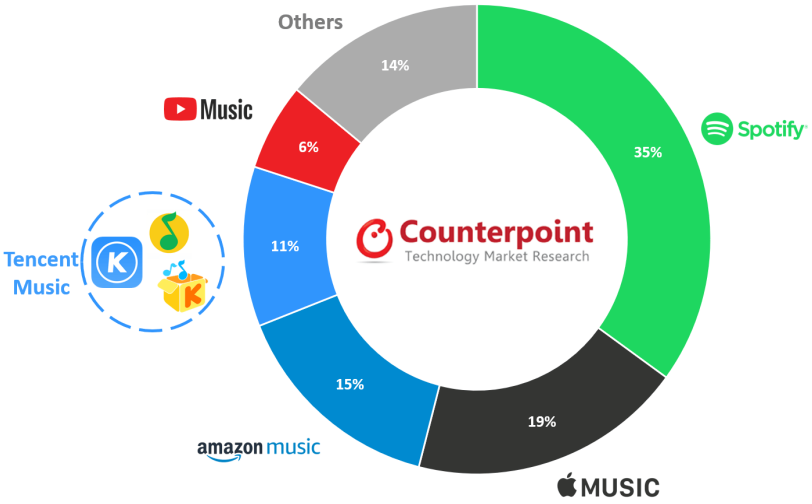
Dla milionów ludzi na całym świecie słuchanie muzyki jest jednoznaczne ze strumieniowym przesyłaniem. Jednak zanim technologia pozwoliła na dotarcie do tego momentu, wyglądało to zupełnie inaczej. Kiedy połączenia internetowe zaczęły się rozwijać m.in. pod względem szybkości, a w ofertach pojawiły się pojemniejsze dyski twarde, udostępnianie muzyki w Internecie (często w sposób nielegalny) stało się bardziej powszechne. Zanim wytwórnie zdążyły dostosować się do powstającego modelu muzyki cyfrowej, pojawiła się aplikacja Napster, która mimo swojego krótkiego istnienia, spowodowała jeden z największych punktów zwrotnych branży muzycznej. Aplikacja umożliwiała użytkownikom dzielenie się plikami muzycznymi MP3, wyszukiwanie ich i pobieranie. Serwis ten jednak został zamknięty i przejęty ze względu na naruszanie praw autorskich. Po Napsterze pojawiło się mnóstwo podobnych witryn, takich jak Rapidshare, Zippyshare, Megaupload, pozwalających na pobieranie plików muzycznych chronionych prawem autorskich oraz Torrentów, które umożliwiają szybsze pobieranie i wysyłanie różnych sekcji plików z wielu źródeł równocześnie. Pierwszą alternatywą dla nielegalnych witryn stał się ITunes Store firmy Apple powstały w 2003 roku. W tym czasie był to jedyny cyfrowy sklep, oferujący katalogi największych wytwórni muzycznych. W późniejszym czasie zaczęły się pojawiać platformy do strumieniowego przesyłania muzyki, tj. YouTube, Spotify, Tidal, czy Apple Music, które pozwoliły odwrócić znaczną część ludzi od stron służących do nielegalnego udostępniania muzyki, nie generujące przychodów dla artystów[1].

Według badań firmy Counterpoint Research Spotify okazał się być najpopularniejszym medium poprzez udział 35% wszystkich płatnych abonentów, stanowiąc tym samym 31% całkowitych przychodów na rynku usług strumieniowego przesyłania muzyki online[2], co można zauważyć na *Rysunku 1*.

[1] *The history of music distribution*, MN2S, 2020, https://mn2s.com/news/label-services/the-history-of-music-distribution/ (dostęp: 12.07.2020)

[2] Majchrzyk Ł., *Spotify i Apple Music rządzą na rynku streamingu muzyki*, Mobirank, 2019, https://mobirank.pl/2020/04/04/spotify-i-apple-music-rzadza-na-rynku-streamingu-muzyki-2019/ (dostęp: 12.07.2020)

Rysunek 1 Procentowy udział muzycznych serwisów streamingowych na świecie pod względem liczby płatnych subskrypcji w 2019 roku.



Źródło: Majchrzyk Ł., *Spotify i Apple Music rządzą na rynku streamingu muzyki*, Mobirank, 2019, https://mobirank.pl/2020/04/04/spotify-i-apple-music-rzadza-na-rynku-streamingu-muzyki-2019/ (dostęp: 12.07.2020)

Tego typu serwisy znacząco zmieniły nie tylko sposób, w jaki słucha się obecnie muzyki oraz zarabia na płytach, ale także umożliwił nieznane dotąd podejście do tworzenia nowych piosenek czy powstawania algorytmów rekomendujących, którym między innymi zawdzięcza swoją popularność. Spotify stara się stosować podejścia, pozwalające na odfiltrowanie niechcianych lub zbędnych informacji na podstawie dotychczasowych zachowań użytkowników i odsłuchanych utworów, a co za tym idzie, umożliwiające odkrywanie przez subskrybentów nowych artystów i albumów, które potencjalnie mogą trafić w ich gust muzyczny. Jest to swego rodzaju technika personalizacji usługi, która obecnie wykorzystywana jest w prawie każdej dziedzinie. Główną funkcją systemu rekomendującego Spotify jest analiza i ekstrakcja danych dotyczących przesłuchanych piosenek, które mogą charakteryzować konkretny gatunek lub okoliczności, w jakich utwór może być odtwarzany, np. podczas nauki, do snu, w czasie podróży itd. W związku z tym dane o utworach wykorzystywane są przy ich klasyfikacji, co jest bardzo istotnym aspektem na etapie opracowywania mechanizmów poleceń utworów. Aby tego typu systemy działały w sposób jak najbardziej dokładny i nowe, nieznane piosenki trafiały do rekomendacji, kluczowe jest między innymi poprawne rozpoznanie podobnych cech charakterystycznych dla danego gatunku lub grupy utworów.

Cel i zawartość analizy

Niniejsza praca ma na celu porównanie wybranych metod klasyfikacji i grupowania, które są stosowane w budowaniu systemów rekomendacji w popularnych muzycznych serwisach streamingowych, takich jak Spotify.

Pierwszy rozdział stanowić będzie przybliżenie działalności serwisu Spotify oraz wyjaśnienie stosowanego w nim działania silnika rekomendacji opartego na trzech metodach: filtrowaniu kooperacyjnym, przetwarzaniu języka naturalnego oraz cyfrowego przetwarzania sygnału. Dodatkowo opisany zostanie interfejs programowania aplikacji, za pomocą którego pozyskane będą dane użyte do analizy.

Kolejny rozdział poświęcony będzie analizie skupień. Opisane zostaną metody grupowania, a w szczególności algorytm *K*-średnich jako przykład metody niehierarchicznej oraz metoda Warda jako przykład metody hierarchicznej, które będą podlegać porównaniu. Wybór optymalnej liczby klastrów oraz przygotowanie danych są ważnymi etapami grupowania, a więc nie zabraknie również przedstawienia zarysów metod, które temu służą.

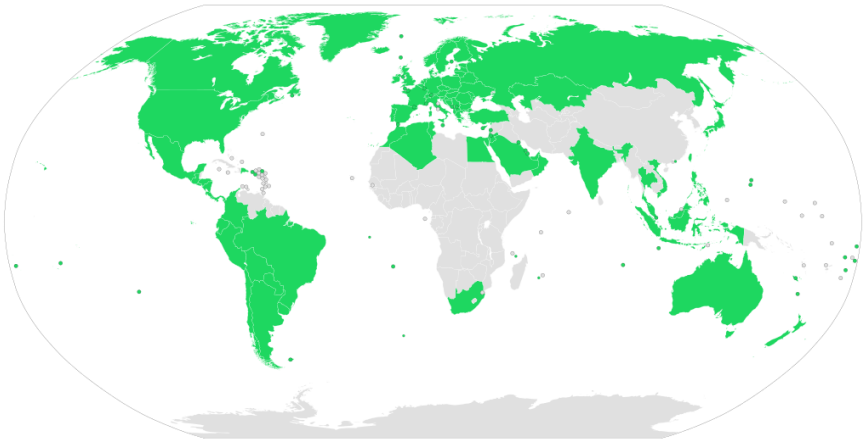
Rozdział trzeci dotyczyć będzie wybranych algorytmów uczenia maszynowego, które użyte zostaną do budowy modeli dokładniejszej klasyfikacji utworów. Bliżej przedstawione zostaną metoda K-najbliższych sąsiadów, drzewa klasyfikacyjne i las losowy oraz sieci neuronowe. Ostatnia część wprowadzi również miary służące ocenie opisanych modeli, w szczególności macierz pomyłek, czułość oraz specyficzność.

Ostatni rozdział zawiera badanie, w którym za pomocą opisanych technik klasyfikacji i grupowania sporządzone zostaną modele z użyciem danych dotyczące cech audio utworów muzycznych wszystkich gatunków pojawiających się w serwisie Spotify. Obejmują one informacje o akustyczności, taneczności, tempie, energiczności, głośności, tonacji, metrum, skali, długości, instrumentalności, ilości słów oraz nastroju utworów. Modele poddane zostaną porównaniu pod kątem dokładności i skuteczności klasyfikacji.

1. Serwis Spotify

Obecnie jeden z najpopularniejszych serwisów do słuchania muzyki został założony przez Daniela Eka oraz Martina Lorentzona w 2006 roku. Za opracowanie oraz rozwój usługi odpowiada Spotify AB z siedzibą w Sztokholmie. Na ten moment ze Spotify mogą skorzystać użytkownicy 79 krajów[3] (*Rysunek 2*).

Rysunek 2 Mapa dostępności Spotify



Źródło: https://pl.wikipedia.org/wiki/Spotify (dostęp: 14.07.2020)

Spotify jest medium strumieniowym, co oznacza, że wykorzystuje techniki dostarczania informacji multimedialnej użytkownikowi w sposób ciągły[4]. W swojej ofercie posiada dostęp do muzyki i podcastów, a także możliwość odkrywania nowej muzyki poprzez listy znajomych, czy rekomendacje. Udostępnianie mediów w tym serwisie opiera się na tzw. licencji freemium, polegającej na darmowym dostępie do produktów, jednak aby skorzystać z bardziej zaawansowanych funkcji konieczne jest wykupienie wersji premium[5]. W wersji bezpłatnej Spotify użytkownik musi liczyć się z emisją reklam w trakcie słuchania muzyki oraz w aplikacji. Dodatkowo wymagane jest stałe połączenie z Internetem, a więc nie jest możliwe pobieranie utworów i odsłuchiwanie ich w dowolnym miejscu w przypadku braku połączenia sieciowego, co znajduje się w ofercie płatnej. Użytkownicy premium mają możliwość słuchania muzyki w dowolnej kolejności oraz pomijania niechcianych pozycji z playlisty bez limitu, co niestety nie występuje w podstawowej wersji serwisu dostępnej dla wszystkich.

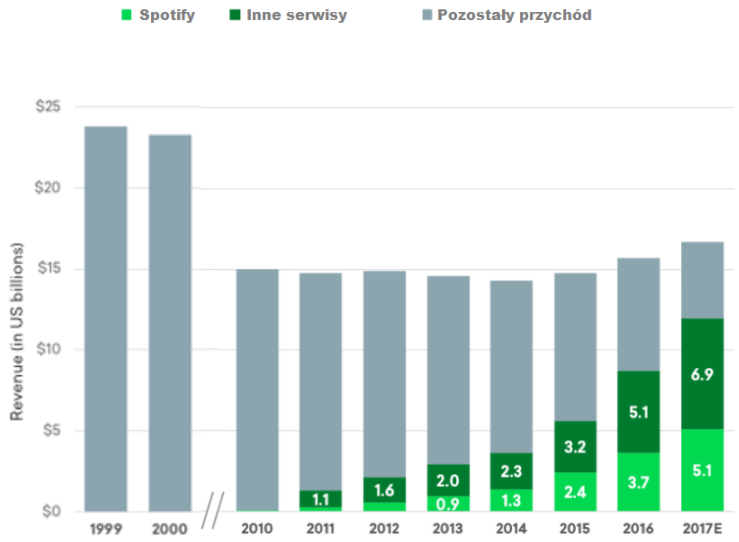
[3] *Spotify*, https://pl.wikipedia.org/wiki/Spotify (dostęp: 12.07.2020)
[4] *Media strumieniowe*, https://pl.wikipedia.org/wiki/Media_strumieniowe (dostęp: 12.07.2020)
[5] *Freemium*, https://pl.wikipedia.org/wiki/Freemium (dostęp: 12.07.2020)

Niewątpliwie różnicą między wersją płatną i bezpłatną jest również jakość odtwarzanej muzyki. Proponowane są 4 opcje kont premium w serwisie: konto indywidualne, dla dwóch osób mieszkających pod tym samym adresem, studenckie ze specjalną zniżką oraz rodzinne do 6 osób[6].

1.1 Rozwój Spotify

Początkowa fala internetowych serwisów muzycznych nie działała na korzyść rynku, ponieważ utwory udostępniane były często nielegalnie. Na przestrzeni lat 1999-2014 doprowadziło to do znacznego spadku globalnych przychodów z muzyki o ok. 40% (*Rysunek 3*).

Rysunek 3 Globalne przychody na rynku muzycznym w latach 1999-2017 (w miliardach USD)



Źródło: *Understanding Spotify: Making Music Through Innovation*, Goodwater, 2018, https://www.goodwatercap.com/thesis/understanding-spotify (dostęp: 14.07.2020)

Model prezentowany przez Spotify ustanowił punkt zwrotny w przemyśle muzycznym poprzez największy roczny przyrost przychodów od kilkunastu lat w 2016 roku oraz widoczny rosnący trend. Co więcej, przychody szwedzkiego serwisu stanowią ponad 30% globalnego przychodu na rynku muzycznym oraz przewyższają 40% sektora związanego ze streamingiem.

Spotify opiera swoją strategię na kontrolowanym zdobywaniu dużej grupy odbiorców poprzez stopniowe rozprzestrzenianie się na świecie, co pozwoliło na wzrost zainteresowania potencjalnych użytkowników ze względu na wyraźny niedobór tego

[6] Oficjalna strona Spotify, https://www.spotify.com/pl/premium/ (dostęp: 12.07.2020)

typu rozwiązań na rynku. Ogromny udział w pozyskiwaniu klientów ma usługa freemium, która bardzo często sprawia, że klient decyduje się skorzystać z rozszerzonej oferty.

Serwis ten zawdzięcza swój sukces temu, iż wykorzystuje społeczny charakter muzyki, aby przyspieszyć udostępnianie i odkrywanie nowości przez użytkowników, a także aby zwiększyć dystrybucję i zasięgi. Jest to przeciwieństwo do wcześniejszego piractwa muzycznego, które działało na niekorzyść w generowaniu przychodów przemysłu muzycznego. Spotify dodatkowo zapewnia twórcom platformę do zarabiania na swojej pracy, nawiązywania kontaktów z fanami i uzyskiwania dostępu do analiz, aby lepiej zrozumieć i rozwijać działalność. Mówiąc o powodzeniu Spotify w dziedzinie muzyki, nie należy również pominąć odpowiedniego momentu wejścia firmy na rynek, kiedy wydajność branży drastycznie spadała w związku z dostępnością nielegalnych źródeł słuchania muzyki. Wydawcy i wytwórnie były otwarte na wypróbowanie nowych modeli przychodów w celu ożywienia wzrostu, a konsumenci byli gotowi przyjąć nową płatną opcję dostępu do nieograniczonej bazy utworów muzycznych. Strategia firmy opiera się również na możliwości personalizacji opartej na danych, która pozwala na odkrywanie nowości na podstawie gustu muzycznego[7].

1.2 Silnik rekomendacji

Branża muzyczna, podobnie jak inne dziedziny, wykorzystuje na co dzień różne techniki uczenia maszynowego, zapewniając swoim odbiorcom spersonalizowane listy odtwarzania. Spotify co tydzień oferuje słuchaczom przystosowaną według gustu playlistę z nowościami muzycznymi, tzw. *Discover Weekly*. Zawiera około 30 utworów do tej pory nie odtworzonych przez użytkownika, wybranych na podstawie odsłuchanych piosenek, które potencjalnie mogą mu spodobać i rozszerzyć spektrum słuchanych artystów.

[7] *Understanding Spotify: Making Music Through Innovation*, Goodwater, 2018, https://www.goodwatercap.com/thesis/understanding-spotify (dostęp: 14.07.2020)

System łączy w swoim działaniu trzy modele w celu analizy podobieństwa utworów:

- filtrowanie kooperacyjne, wyodrębniające i identyfikujące utwory na listach odtwarzania, których użytkownik nie słyszał w Spotify na podstawie gustu innych użytkowników,
- przetwarzanie języka naturalnego, służące do analizy tekstu w utworach,
- cyfrowe przetwarzanie sygnału, które ma na celu zrozumienie utworu na podstawie analizy dźwięku.

Filtrowanie kooperacyjne

Algorytm filtrowania kooperacyjnego (ang. *colaborative filtering*) wykorzystywany przez Spotify polega na przewidywaniu gustu słuchacza i określeniu wzorca na podstawie zebranych preferencji innych podobnych użytkowników. Sposób na określanie podobieństwa użytkowników opiera się na sprawdzeniu, czy analizowana osoba słuchała tych samych utworów, co inny użytkownik[8]. Na początku zbierane są informacje ze wszystkich istniejących list odtwarzania o utworach i artystach lubianych przez słuchacza, a następnie identyfikowane. Ostatecznie filtrowane są tylko utwory, których użytkownik nie słyszał na Spotify. Platforma Spotify nie używa oceniania utworów w formie przyznawanych gwiazdek, jak niektóre serwisy streamingowe. W przeciwieństwie do nich, informacja zwrotna opiera się na interakcji użytkownika z systemem na zasadzie zliczania utworów, które zostały odsłuchane lub dodane do listy odtwarzania, a także odwiedzin strony artysty[9].

Algorytmy filtrowania kooperacyjnego CF dzielą się na nieprobabilistyczne, czyli nie wykorzystujące losowości i probabilistyczne, używające w obliczeniach generatora liczb losowych. Najbardziej znaną techniką nieprobabilistyczną CF jest algorytm *K*-najbliższych sąsiadów, w których prognozy dla użytkowników generowane są na podstawie ocen podobnych użytkowników, czyli sąsiadów. Algorytm ten zostanie opisany bardziej szczegółowo w *Rozdziale 3*. Równie często stosowane są metody oparte na grafach, sieci neuronowe i algorytmy eksploatacji reguł.

[8] Giacaglia G., *Spotify's Recommendation Engine*, Medium, 2019, https://medium.com/datadriveninvestor/behind-spotify-recommendation-engine-a9b5a27a935 (dostęp: 15.07.2020)

[9] Guneysu U., *How Is Spotify's Thriving Recommendation System Becoming A New Advertising Platform*, Medium, 2019, https://medium.com/swlh/how-is-spotifys-thriving-recommendation-system-becoming-a-new-advertising-platform-a2b97ffe2012 (dostęp: 15.07.2020)

Probabilistyczne algorytmy reprezentują rozkłady prawdopodobieństwa podczas obliczania przewidywanych ocen lub sporządzanie rankingowych list rekomendacji. Najpopularniejszymi z nich są sieci bayesowskie z użyciem drzew decyzyjnych, które generują zależności probabilistyczne między użytkownikami lub elementami[10].

Przetwarzanie języka naturalnego

Przetwarzanie języka naturalnego (ang. *Natural Language Processing*) jest dziedziną sztucznej inteligencji, która koncentruje się na zrozumieniu języka ludzkiego przez maszynę. W zastosowaniu Spotify polega on na skanowaniu metadanych utworów, blogów, wiadomości oraz mediów społecznościowych w poszukiwaniu słów kluczowych powiązanych z piosenkami i artystami i tego, co ludzie piszą na ich temat. Taka analiza pozwala na określenie przymiotników i języka używanych do opisania utworów i ich wykonawców, a następnie na wybranie najczęściej pojawiających się słów w tego typu opisach. Każde z nich ma przypisaną wagę, która oznacza jak istotny jest dany opis dla piosenki lub artysty. Ostatecznie tagi i słowa dodawane są do modelu każdego utworu i wykonawcy, aby później mogły zostać wykorzystane w wyszukiwaniu polecanych utworów dla użytkownika.

Cyfrowe przetwarzanie sygnału w modelowaniu dźwięku

Trzeci wykorzystywany model bazujący na analizie i przetworzeniu zapisu dźwięku (ang. *Digital Signal Processing*) pozwala nie tylko na zwiększenie dokładności systemu rekomendacji Spotify, ale też odgrywa ważną rolę w wyszukiwaniu nowych piosenek. Ma to na celu niedopuszczenie do momentu, w którym algorytm wybiera stale te same, znane utwory, ale bierze też pod uwagę też mniej znane pozycje, o których nie ma zbyt wielu informacji. Metoda ta używa surowego zapisu dźwięku, tzw. RAW, wyodrębniającego najczęściej kilka ścieżek i kanałów, na których zapisane są poszczególne instrumenty i wokal, co jest wykorzystywane do analizy pod kątem podobieństwa pomiędzy utworami. Dzięki temu każdy utwór niezależnie od tego jak bardzo jest znany i jak długo istnieje na rynku muzycznym ma równe szanse na znalezienie się na rekomendowanej liście odtwarzania.

[10] Schafer J.B. , Frankowski D., Herlocker J , Sen S., *Collaborative Filtering Recommender Systems,* Conference Paper on ResearchGate, 2007, https://www.researchgate.net/publication/200121027_Collaborative_Filtering_Recommender_Systems (dostęp 15.07.2020)

Do przeprocesowania surowego zapisu dźwięku wykorzystywane są konwolucyjne sieci neuronowe, znajdujące często zastosowanie w analizie obrazów z powodu dobrego radzenia sobie z rozpoznawaniem wzorców. Ze względu na stosowane filtry badające zależności pomiędzy sąsiednimi pikselami obrazów, posiadają znaczącą przewagę nad klasycznymi sieciami neuronowymi przy wykrywaniu skomplikowanych zależności. W wyniku przetworzenia sygnału wyodrębniane zostają cechy tj. sygnatura czasowa (*time signature*), tonacja (*key*), tempo, głośność (*loudness*), skala (*mode*) itp. Po przetworzeniu takich danych przez sieci neuronowe możliwe jest pogrupowanie podobnych piosenek i zarekomendowanie ich według gustu słuchacza na podstawie powstałych metryk [11].

Dzięki połączeniu trzech modeli, Spotify ma możliwość przeanalizowania podobieństwa utworów i artystów muzycznych oraz zarekomendowanie nowych, nieznanych wcześniej użytkownikowi piosenek w każdym tygodniu, co zdecydowanie skutkuje popularnością usługi Spotify, jaką jest polecona lista odtwarzania Discover Weekly.

1.3 Spotify Web API

Web API to rodzaj sieciowego interfejsu programowania aplikacji (ang. *Web Application Programming Interfaces*), który do komunikacji między aplikacjami w sieci wykorzystuje architekturę i protokoły sieci Web, najczęściej HTTP. Jest to protokół sieci WWW umożliwiający stronom publikowanie informacji poprzez przesyłanie żądania udostępniania dokumentów i informacji z formularzy. Interfejs webowy pozwala na sprawną wymianę informacji z zewnętrznymi serwisami, nie wymagając do tego tworzenia widoków[12].
Bazując na prostych zasadach REST, Spotify Web API zwraca metadane w formacie JSON dotyczące artystów, albumów i utworów bezpośrednio z katalogu danych Spotify. Interfejs zapewnia również dostęp do danych związanych z użytkownikiem, takich jak listy odtwarzania i muzyka, które użytkownik zapisuje w bibliotece *Your*

[11] Giacaglia G., *Spotify's Recommendation Engine*, Medium, 2019, https://medium.com/datadriveninvestor/behind-spotify-recommendation-engine-a9b5a27a935 (dostęp: 15.07.2020)
[12] *Web API*, https://pl.wikipedia.org/wiki/WebAPI (dostęp: 16.07.2020)

Music, jednak aby uzyskać dostęp do prywatnych danych za pomocą API sieciowego konieczne jest pozwolenie użytkownika za pośrednictwem konta Spotify.

REST

Jak wspomniano, Spotify Web Api opiera się na zasadach REST, co jest skrótem od Representational State Transfer oznaczającym zmianę stanu poprzez reprezentację. Jest to styl architektury oprogramowania, który bazuje na zbiorze określonych reguł definiowania zasobów danych.
Komunikacja między serwerem a klientem odbywa się za pośrednictwem standardowych żądań HTTP lub szyfrowanej wersji HTTPS. Podstawowe API wykorzystuje zazwyczaj cztery metody z dziewięciu, mówiące o rodzaju akcji, jaka zostanie wykonana:

- *GET*- odczyt i pobranie danych,
- *POST*- zapis i tworzenie nowego zasobu,
- *PUT*- aktualizacja poprzez zastąpienie istniejącego zasobu,
- *DELETE* - usuwanie istniejących danych.

Adres zapytania przedstawia zasób, na którym będą wykonywane operacje. Odpowiedź z serwera zwraca kod HTTP, informujący o poprawności zapytania i jego rezultacie. Istotną informacją na temat REST jest to, iż komunikacja jest bezstanowa. Oznacza to, że każde zapytanie jest niezależne, nie istnieją żadne sesje, ani nie wykorzystuje się ciasteczek.

Format danych JSON

JSON, czyli JavaScript Object Notation jest obecnie jednym z najpopularniejszych formatów danych bazującym na podzbiorze języka JavaScript. Używany jest on do komunikacji między serwisami, w plikach konfiguracyjnych bibliotek, czy przy zapisie danych z baz danych. Jego zaletą jest to, iż jest bardzo lekkim formatem wymiany zasobów, który dzięki temu, że poza danymi które przesyła nie zawiera wielu dodatkowych znaków staje się bardzo czytelny. Obiekty w JSON są nieuporządkowanym zbiorem par nazw i wartości oddzielonych przecinkami oraz umieszczonych w nawiasach klamrowych.

Spotify URI i ID

W żądaniach do internetowego interfejsu Spotify i ich odpowiedziach występują różnego rodzaju identyfikatory, które zostały zawarte w *Tabeli 1*.

Tabela 1 Lista parametrów URI i ID Spotify

Parametr	Opis	Przykład
Spotify URI	Identyfikator zasobu, który może być wprowadzony np. w polu wyszukiwania w Spotify Desktop w celu znalezienia wykonawcy, albumu lub utworu.	spotify:track:6rqhFgbbKwnb9MLmUQDhG6
Spotify ID	Identyfikator base-62, który można znaleźć na końcu identyfikatora URI Spotify dla wykonawcy, utworu, albumu, listy odtwarzania itp. W przeciwieństwie do URI, identyfikator nie identyfikuje jasno typu zasobu.	6rqhFgbbKwnb9MLmUQDhG6
Spotify category ID	Unikalny ciąg znaków identyfikujący kategorie.	party
Spotify user ID	Unikalny ciąg znaków identyfikujący użytkownika Spotify, zawarty w URI Spotify dla użytkownika. Identyfikator bieżącego użytkownika można uzyskać za pośrednictwem punktu wyjścia Web API.	wizzler
Spotify URL	Łącze HTML, które pozwala na otwarcie utworu, albumu, aplikacji, listy odtwarzania lub innych zasobów Spotify.	http://open.spotify.com/track/6rqhFgbbKwnb9MLmUQDhG6

Źródło: *Spotify for developers: Web API*, Spotify AB, 2020, https://developer.spotify.com/documentation/web-api/ (dostęp: 16.07.2020)

Można się zastanawiać dlaczego firmy udostępniają swoje API za darmo. Powodów jest wiele, ale są to przede wszystkim kwestie darmowego marketingu i reklamy, a co za tym idzie zachęcenie programistów do proponowania innowacyjnych rozwiązań, zarabianie na danych czy pozyskiwanie partnerów. Skuteczny interfejs może dać obecnym i potencjalnym klientom powody do interakcji z firmą oraz do dzielenia się swoimi doświadczeniami z innymi.

2. Analiza skupień

Wielowymiarowość jest nieodłączną cechą wielu danych statystycznych, dlatego analiza zbiorów obserwacji wymaga zastosowania pewnych metod służących uproszczeniu i zrozumienia ich bez jednoczesnego pominięcia ich złożoności. Bardzo często wartości jednej cechy są powiązane z wartościami innych cech, ale zdarzają się sytuacje w których nie można stwierdzić zależności. Z tego powodu głównymi zastosowaniami analizy wielowymiarowej jest badanie zależności między jednostkami oraz badanie ich łącznej zmienności.

Według *Słownika Terminów Statystycznych* Bucklanda i Kendalla do metod analizy wielowymiarowej należą: analiza głównych składowych, klasyfikacja i analiza skupień, analiza czynnikowa, analiza dyskryminacyjna, analiza kanoniczna, uogólnienie testów jednorodności. Dalsza część pracy będzie stanowić dogłębniejsze spojrzenie na problem klasyfikacji, używany do grupowania podobnych obiektów i organizowania dużych zbiorów danych tak, aby pozyskiwanie informacji było efektywne. Dzięki analizie zróżnicowania badanych obiektów przy użyciu etykiet ich klas, możliwe jest otrzymanie przejrzystego sumarycznego opisu danych[13].

2.1 Idea grupowania i jego etapy

Pojęcie analizy skupień, nazywanej inaczej grupowaniem lub klasteryzacją wywodzi się z szerszej dziedziny, jaką jest klasyfikacja bez nadzoru, która polega na odkrywaniu w zbiorze danych wzorców bez wcześniej istniejących etykiet. Służy do grupowania *n*-obiektów, które są opisywane za pomocą wektora p cech, w *K* rozłącznych i jak najbardziej jednorodnych grup. Główna idea klasteryzacji opiera się na znalezieniu najbardziej podobnych obserwacji należących do tego samego skupienia, ale jednocześnie niepodobnych do obiektów z innych grup[14].

Do zastosowań analizy skupień należą między innymi wstępna analiza danych prowadząca do wyodrębnienia jednorodnych grup i segmentacji, eksploaracja danych oraz wyszukiwanie i uporządkowanie informacji.

Pierwszym etapem analizy skupień jest wybór obiektów i zmiennych. Istotne jest odpowiednie przygotowanie danych poprzez przeprowadzenie normalizacji wartości

[13] Balicki A., *Statystyczna analiza wielowymiarowa i jej zastosowania społeczno-ekonomiczne,* Wydawnictwo Uniwersytetu Gdańskiego, 2013, s. 15-20, 205

[14] Górecki T., Krzyśko M., Skorzybut M., Wołyński W., *Systemy uczące się,* Wydawnictwo Naukowo-Techniczne Warszawa, 2008, s. 345

zmiennych, tak aby było możliwe porównywanie obiektów o nieporównywalnych jednostkach czy skalach. Następnymi krokami są wybór miary odległości, metody klasyfikacji oraz optymalnej liczby klastrów. W efekcie możliwe jest przeprowadzenie oceny wyników klasyfikacji oraz profilowania otrzymanych grup.

2.2 Etap przygotowania danych

Normalizacja zmiennych jest etapem przygotowywania danych polegającym na przypisaniu pierwotnym zmiennym ich przetransformowanych wersji. Do jej najważniejszych celów należy doprowadzenie zmiennych do porównywalności, ujednolicenie ich charakteru, wykluczenie wartości ujemnych oraz stabilizacja zmienności[15]. Ogólny wzór na normalizacje jest następujący:

$$z_{ij} = b_j x_{ij} + a_j = \frac{x_{ij} - A_j}{B_j} = \frac{1}{B_j} x_{ij} - \frac{A_j}{B_j} \qquad (b_j > 0) \qquad (6)$$

gdzie x_{ij} to wartość *j*-tej zmiennej dla *i*-tego obiektu, a z_{ij} to jej znormalizowana wartość. A_j jest parametrem przesunięcia do umownego zera, natomiast B_j parametrem skali. Oznaczenia a_j oraz b_j stanowią parametry dla j-tej zmiennej, przy czym $a_j = -A_j/B_j$ oraz $b_j = 1/B_j$[16].

Najpowszechniejszą metodą normalizacji jest standaryzacja, prowadząca do wyrażenia wartości zmiennej z wykorzystaniem odchylenia standardowego od jej średniej. Dzięki temu różnoimienne zmienne mogą być porównywane w uwagi na względne położenie w rozkładzie (średnia równa zero i wariancja równa 1).

Wyraża się ją wzorem:

$$z_{ij} = \frac{x_{ij} - \overline{x_j}}{s_j} \qquad i = 1, \ldots, n; j = 1, \ldots, p \qquad (7)$$

Stosowane są również inne metody normalizacji, tj. przekształcenie ilorazowe, unitaryzacja czy rangowanie.

[15] Gatnar E., Walesiak M., *Statystyczna analiza danych z wykorzystaniem program R*, Wydawnictwo naukowe PWN, Warszawa, 2012, s.255-256,407-420

[16] Walesiak M., *Przegląd formuł normalizacji wartości zmiennych oraz ich własności w statystycznej analizie wielowymiarowej*, Przegląd statystyczny R. LXI - Zeszyt 4, 2014, s. 364

2.3 Podobieństwo obiektów i miary odległości

Pojęcia odległości i podobieństwa obserwacji odgrywają bardzo ważną rolę w analizie skupień. Miary te muszą być oszacowane zgodnie z zaobserwowanymi wartościami cech. Wynik klasteryzacji jest zależny od tego jaka metoda obliczenia odległości zostanie zastosowana oraz w jaki sposób zostanie zdefiniowana odległość obserwacji od klasy. Podstawą analizy skupień jest weryfikacja stopnia zróżnicowania lub podobieństwa obserwacji w wielowymiarowej przestrzeni cech. Obiekty są takie same w momencie, gdy wektory wartości ich cech są identyczne tj. $x_i = x_k$ dla $i, k = 1,2,\dots,n$ oraz $i \neq k$, jednak realnie trudno osiągnąć taką sytuację. W takim przypadku można mówić jedynie o zbieżności cech.

Miary odległości wykorzystywane do określania podobieństwa obiektów są funkcjami wartości cech $X_1 \dots X_p$ opartymi na dystansach między punktami r i s w przestrzeni p- wymiarowej:

$$d_{rs} = f(x_r, x_s) \tag{1}$$

Odległość dwóch punktów to wartość metryki o następujących własnościach:

- nieujemność odległości, równoznaczna z rozróżnianiem obiektów nieidentycznych: $d_{rs} \geq 0$ dla $x_r \neq x_s$ i $(r, s = 1, \dots, n)$
- nierozróżnianie obiektów identycznych: $d_{rs} = 0 \leftrightarrow x_r = x_s$ i $(r, s = 1, \dots, n)$
- symetria w odległości dwóch obiektów: $d_{rs} = d_{sr}$ dla wszystkich $(r, s = 1, \dots, n)$
- suma odległości między obiektami r i x oraz s i x jest nie mniejsza niż odległość między obiektami r i s (tzw. nierówność trójkąta): $d_{rs} \leq d_{rx} + d_{sx}$ dla $r, s, x = 1, \dots, n$.

Najczęściej stosowane miary odległości bazują na metryce potęgowej (2), w której wagi dużych i małych różnic określane są przez wykładnik potęgi definiowany przez stałą Minkowskiego.

$$d_{rs}^{(m)} = \left[\sum_{j=1}^{p} |x_{rj} - x_{sj}|^m \right]^{\frac{1}{m}} \tag{2}$$

W szczególności należy wyróżnić miejską (m=1) i metrykę euklidesową (m=2). Odległość euklidesowa określona wzorem (3) jest najbardziej klasyczną miarą. Często w celu uwydatnienia małych różnic między obiektami stosowana jest wersja z kwadratem tej metryki. Inną modyfikacją w przypadku, gdy cechy opisujące obiekty podane są w różnych jednostkach, jest postać ważona mająca na celu zniwelowanie ich wpływu.

$$d_{rs} = \sqrt{\sum_{j=1}^{p} (x_{rj} - x_{sj})^2} \tag{3}$$

Odległość miejska, definiowana wzorem (4), w odróżnieniu od metody euklidesowej nie jest czuła na duże odległości między obiektami.

$$d_{rs} = \sum_{j=1}^{p} |x_{rj} - x_{sj}| \tag{4}$$

Czasami stosowana jest również odległość Czebyszewa, która wybiera największą odległość między obiektami:

$$d_{rs} = max_j |x_{rj} - x_{sj}| \tag{5}$$

Wartości mierników odległości przedstawiane są w formie symetrycznej macierzy *n* x *n* odległości[17]:

$$D = d_{rs} = \begin{bmatrix} 0 & d_{12} & \dots & d_{1n} \\ d_{21} & 0 & \dots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots & 0 \end{bmatrix} \tag{6}$$

Korzystanie z wymienionych metryk ma sens tylko w przypadku porównywalnych zmiennych, dlatego wskazane jest przeprowadzenie normalizacji lub standaryzacji.

[17] Balicki A., *Statystyczna analiza wielowymiarowa i jej zastosowania społeczno-ekonomiczne*, Wydawnictwo Uniwersytetu Gdańskiego, 2013, s. 214-222

2.4 Metody grupowania

W związku z występowaniem różnych podejść do określania podobieństwa obiektów w zbiorze oraz przetwarzania danych istnieje kilka grup algorytmów służących do klasteryzacji:

- hierarchiczne (aglomeracyjne i deglomeracyjne),
- podziałowe, optymalizacyjno-iteracyjne (K-średnich, K-mediana),
- inne algorytmy oparte o m.in. sieci neuronowe, algorytmy genetyczne, teorię grafów[18].

Bliżej omówione zostaną metody z grupy K-średnich oraz hierarchiczne.

Metody hierarchiczne

Grupa algorytmów hierarchicznych ma na celu tworzenie hierarchii klasyfikacji dla zbioru obiektów w oparciu o podobieństwo. Strategie tego typu grupowania dzielą się na dwa podejścia ze względu na sposób konstruowania hierarchii: aglomeracyjne i deglomeracyjne.

W algorytmach hierarchicznych aglomeracyjnych na początku wszystkie elementy ze zbioru stanowią samodzielne skupienia. Obiekty te są sekwencyjnie grupowane poprzez łączenie ich w skupienia najbardziej zbliżonych do siebie obiektów na podstawie obliczonych miar odległości i podobieństwa. Skupienie dwóch najbardziej podobnych do siebie obiektów prowadzi do wyodrębnienia pierwszej grupy, z kolei następne obiekty wpływają na połączenie grup bądź są przydzielane do istniejącego skupienia. W każdym kroku liczba skupień zmniejsza się o 1, co oznacza, że łączone są tylko dwa skupienia na określonym poziomie, ustalonym poprzez największe podobieństwo lub najmniejszą odległość. Każde powtórzenie wymaga sporządzenia nowej macierzy odległości, czyli przeliczenia odległości między istniejącymi i nowo powstałymi klastrami, ponieważ na ich podstawie tworzone będą nowe grupy. Obliczenia wykonywane są do momentu osiągnięcia ustalonej końcowej liczby skupień lub połączenia wszystkich elementów w jedną grupę. W efekcie pozwala to na otrzymanie ostatecznej grupy obiektów i dendrogramu, który jest graficzną reprezentacją cech

[18] Mazur D., *Metody grupowania i ich implementacja do eksploracji danych postaci symbolicznej*, Praca doktorska, 2005

wykorzystanych do hierarchicznego uporządkowania. Schemat łączenia klas definiuje poziomy skupień, które oznaczają poszczególne podziały na grupy[19].

Proces grupowania w metodach deglomeracyjnych przebiega w odwrotny sposób poprzez rozpoczęcie od zbioru łączącego wszystkie obiekty oraz dokonywanie sekwencyjnego podziału aż do uzyskania ustalonej liczby podzbiorów. Problemem tej techniki jest trudność w określeniu kryterium podziału, ponieważ konieczne jest rozpatrzenie różnych opcji podziału przez uwzględnienie wszystkich atrybutów obiektów. Dlatego często algorytmy należące do tej grupy są bardziej złożone obliczeniowo i dają gorsze rezultaty niż aglomeracyjne[20].

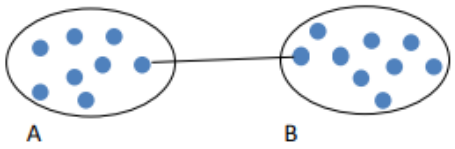
Poza miarami niepodobieństwa obiektów algorytmy hierarchiczne stosują różne techniki łączenia skupień służące do definiowania odległości między klastrami. Należą do nich następujące metody:

a) pojedynczego wiązania (*single*) – najbliższego sąsiedztwa,
b) pełnego wiązania (*complex*) – najdalszego sąsiedztwa,
c) średnich połączeń (*average*) – średniej grupowej,
d) centroidalna (*centroid*) – środków ciężkości,
e) mediany (*median*) – ważonych środków ciężkości,
f) Warda.

Metoda pojedynczego wiązania oparta jest na obliczaniu odległości między obiektami różnych skupień według kryterium najmniejszej odległości. W związku z tym odległości pomiędzy nową grupą, a już istniejącymi definiowane są przez najbliższe elementy należące do różnych skupień, jak przedstawiono na *Rysunku 4*. Jedną z podstawowych własności tej metody jest to, że skupienia stają się nadmiernie rozproszone, a więc słabo radzi sobie ona z odróżnieniem mało rozdzielonych grup. Prowadzi do tworzenia się tzw. łańcuchów, czyli wydłużonych skupień powstałych na skutek pośredniczenia najbliższych jednostek między kolejnymi grupami. Ten typ wiązania jest określany jako zmniejszający odległości.

[19] Balicki A., *Statystyczna analiza wielowymiarowa i jej zastosowania społeczno-ekonomiczne,* Wydawnictwo Uniwersytetu Gdańskiego, 2013, s. 259-261
[20] Mazur D., *Metody grupowania i ich implementacja do eksploracji danych postaci symbolicznej*, Praca doktorska, 2005

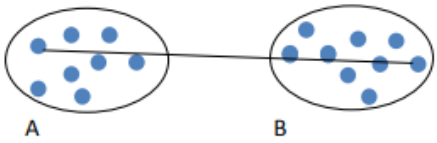
Rysunek 4 Metoda pojedynczego wiązania



Źródło: *Analiza danych pomiarowych: Analiza skupień*, AGH Inżynieria biomedyczna, 2014/2015, http://home.agh.edu.pl/~mmd/_media/dydaktyka/adp/analiza_skupien.pdf (dostęp: 26.07.2020)

Ze względu na ograniczenia metody najbliższego sąsiada stosowana jest metoda pełnego wiązania. Podejście to opiera się na minimalizacji największych międzygrupowych odległości. Innymi słowy, wybierane są najmniejsze odległości, które w odróżnieniu do poprzedniej metody zostały obliczone w oparciu o najdalsze obiekty, co widać na *Rysunku 5*. Strategia najdalszego sąsiada ma tendencję do znajdywania grup bardziej spójnych i zwartych. Nazywana jest ona metodą wydłużającą odległości.

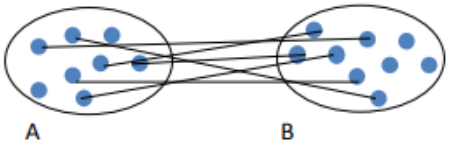
Rysunek 5 Metoda pełnego wiązania



Źródło: *Ibidem*

Jako, że metody te nie są idealne ze względu na wrażliwość na wartości ekstremalne, stosowane jest czasami rozwiązanie pośrednie, nazywane średnim połączeniem. Bazuje ono na średniej odległości, będącej średnią arytmetyczną odległości par należących do porównywanych skupień (*Rysunek 6*). Wadą tej metody jest brak zdefiniowanych centrów. Istnieje również metoda średnich połączeń ważonych, która uwzględnia wielkości skupień, dzięki czemu znajduje zastosowanie w przypadku wyraźnej różnicy w licznościach grup.

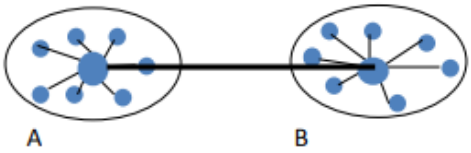
Rysunek 6 Metoda średnich połączeń



Źródło: *Ibidem*

Istotą metody centroidalnej z kolei jest określenie odległości między skupieniami z użyciem punktów ciężkości o współrzędnych będących średnimi wartościami cech wszystkich elementów w poszczególnych skupieniach (*Rysunek 7*).

Rysunek 7 Metoda centroidalna



Źródło: *Ibidem*

Podobnie jak poprzednia procedura, technika środków ciężkości ma swój ważony odpowiednik uniezależniający wynik od liczebności grup. Określana jest metodą mediany. Podsiada ona jednak tę samą wadę co metoda najbliższego sąsiada poprzez wykazywanie tendencji do łączenia się obiektów na względnie niskim poziomie przez jednostki pośredniczące, utrudniając w ten sposób wyodrębnienie skupień.

Ostatnie z podejść, nazywane metodą Warda różni się od pozostałych pod względem kryterium łączenia skupień oraz pomiaru podobieństwa. Algorytm ten opiera się na analizie wariancji. Ideą jest optymalizacja podziału powstałego w wyniku skupienia dwóch obiektów poprzez minimalizację przyrostu łącznej sumy kwadratów odchyleń standardowych zmiennych od ich średnich grupowych wewnątrz skupienia. W efekcie na każdym etapie wybiera tę parę, która tworzy skupienie o najmniejszym zróżnicowaniu. Miarą zróżnicowania skupienia względem wartości średniej w obrębie każdego skupienia jest suma kwadratów błędów wyrażona wzorem (7).

$$W_g^{(k)} = \sum_{i=1}^{g} (x_{ij} - \bar{x}_j)^2 \qquad g = 1,\ldots,n-k \tag{7}$$

Suma ta jest obliczana na każdym etapie grupowania k (gdzie k = 1,…,n-1) dla n-k występujących skupień. Łączna wewnątrzgrupowa suma kwadratów obliczana jest z

kolei poprzez zsumowanie po skupieniach i powinna być obliczana na każdym etapie łączenia. W kolejnych krokach sukcesywnie zmniejsza się liczba grup przy jednoczesnym powiększaniu liczebności każdej z nich.

Metoda Warda ma tendencję do tworzenia skupień o podobnej liczności, dlatego w przypadku gdy dany punkt jest prawie jednakowo oddalony od centroidów dwóch skupień o różnej liczbie elementów, obiekt ten trafi do grupy mniej licznej. Jest to jednoznaczne z szybszym przyłączaniem się nowych obiektów do mniejszych skupień, a co za tym idzie, uniknięciem tworzenia łańcuchów[21].

Metoda *K*-średnich

Jedną z najpopularniejszych metod podziałowych wykorzystywaną w analizie skupień jest *K*-średnich, reprezentująca grupę algorytmów niehierarchicznych. Podejście to polega na podziale zbioru danych na z góry określoną liczbę rozłącznych grup, określaną dalej jako *K*. Wybór początkowej liczby klastrów dokonywany jest losowo, arbitralnie lub według ustalonych kryteriów, które zostaną opisane w kolejnym podrozdziale. Często stanowi to odrębną częścią procesu grupowania[22].

Grupowanie podziałowe ma na celu znalezienie takiego podziału, w którym podzbiory będą równocześnie homogeniczne i dobrze oddzielone, czyli minimalizujące zmienność wewnątrz skupienia oraz maksymalizujące zmienność między grupami.

Algorytm *K*- średnich przebiega w trzech podstawowych krokach:

a) Każdemu z n obiektów ze zbioru w sposób losowy przypisywany jest numer klastra od 1 do *K*, co stanowi początkowe rozmieszczenie elementów określone funkcją C_K.
b) Dla każdego skupienia wyznaczane są centroidy, czyli wektor średnich $\bar{x}_k$, gdzie $\mathrm{k} = 1,2,\dots,\mathrm{K}$.
c) Obiekty są ponownie rozmieszczane do najbliższego centroidu, zgodnie z zasadą zawartą we wzorze (8), wykorzystującą kwadrat odległości euklidesowej.

$$C_K^{(l)}(\mathrm{i}) = \arg\min ||x_i, \bar{x}_k||^2 \qquad k = 1,\dots,K \tag{8}$$

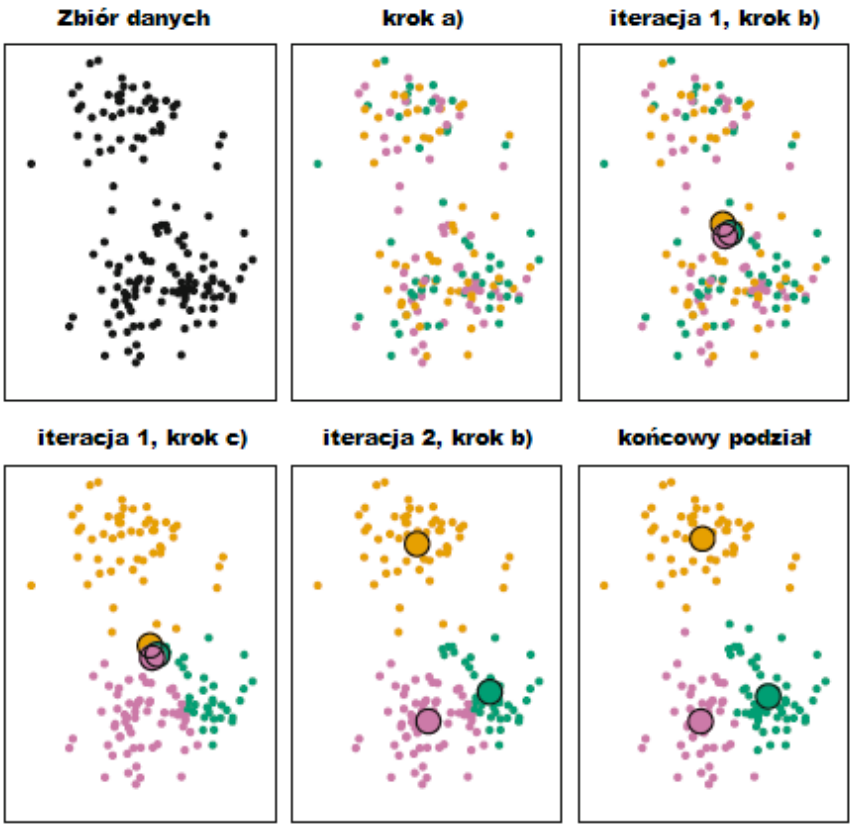
[21] Balicki A., *Statystyczna analiza wielowymiarowa i jej zastosowania społeczno-ekonomiczne,* Wydawnictwo Uniwersytetu Gdańskiego, 2013, s. 259-279

[22] *Analiza danych pomiarowych: Analiza skupień*, AGH Inżynieria biomedyczna, 2014/2015, http://home.agh.edu.pl/~mmd/_media/dydaktyka/adp/analiza_skupien.pdf (dostęp: 26.07.2020)

Kroki b) i c) są powtarzane aż do momentu, gdy przydzielenie obiektów do skupień pozostanie bez zmian, tzn. gdy $C_K^{(l)} = C_K^{(l-1)}$[23].

Elementy procesu przedstawione zostały w uproszczony sposób na *Rysunku 8*.

Rysunek 8 Wizualizacja działania algorytmu k-średnich



Źródło: Hastie T., James G., Witten D., Tibshirani R., *An Introduction to Statistical Learning*, Springer, New York, 2013, s. 389

[23] T.Hastie, R.Tibshirani, J.Friedman, *The Elements of Statistical Learning. Data Mining, Inference, and Prediction. Second Edition*, Springer, 2009, s. 509-510

2.5 Metody wyboru optymalnej liczby skupień

Istotnym etapem w analizie skupień i stosowaniu algorytmów jest określenie liczby klastrów, która będzie brana pod uwagę w grupowaniu. Wybór liczby grup często jest niejednoznaczny, dlatego najlepszym rozwiązaniem jest znalezienie optymalnej liczby, która pozwoli na maksymalną dokładność grupowania.

Podstawową metodą do wyboru odpowiedniej liczby klas stosowaną w metodach hierarchicznych jest analiza dendrogramu, ilustrującego przebieg aglomeracji. Polega ona na ustaleniu progu miary niepodobieństwa między grupami, a następnie przerwaniu procesu klastrowania po przekroczeniu tej wartości [24].

W przypadku grupowania niehierarchicznego kwestia liczby grup jest poważniejszym problemem, ponieważ poszczególne partycje będące kolejnym szczeblem hierarchii nie są widoczne w tak łatwy sposób. Jako, że wybór odpowiedniej liczby klas jest ważną kwestią w większości zastosowań optymalizacyjnych metod klasyfikacji, stosowany jest szereg indeksów ułatwiających tę decyzję. Poniżej opisane zostaną wybrane metody.

Metoda łokciowa

Wiąże się ona z ideą stosowaną w większości metod klasyfikacji, czyli wyborze klastrów tak, aby łączna wariancja była minimalna. Wariancja ta jest sumą kwadratów odległości pomiędzy każdą z obserwacji, a środkiem klastra:

$$\min \left(\sum_{i=1}^{k} W(C_i) \right) \tag{9}$$

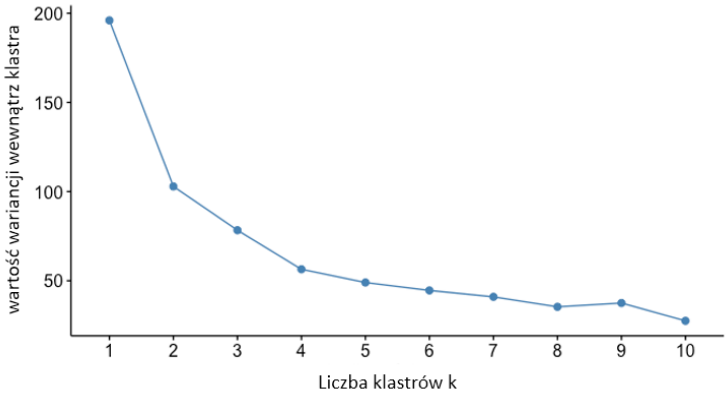
gdzie C_i to i-ty klaster, a $W(C_i)$ to wartość wariancji wewnątrz klastra.

Wraz ze wzrostem podziału na większą liczbę klastrów suma wariancji będzie malała, co się dzieje z powodu zmniejszania się grup. Poszukiwana optymalna liczba klas znajduje się więc w miejscu, gdzie suma wariancji przestaje gwałtownie spadać na tzw. łokciu, ponieważ wtedy dokładanie kolejnych klastrów niewiele zmienia[25]. Zależność tę najłatwiej zaobserwować na wykresie jak na *Rysunku 9,* gdzie wspomniany łokieć jest widoczny już przy 4 klastrach.

[24] Górecki T., Krzyśko M., Skorzybut M., Wołyński W., *Systemy uczące się*, Wydawnictwo Naukowo-Techniczne Warszawa, 2008, s.382

[25] *Optymalna liczba klastrów w metodzie k-średnich w R,* Quant blog, 2016, http://quantblog.pl/2016/04/01/optymalna-liczba-klastrow-w-metodzie-k-srednich/ (dostęp: 25.07.2020)

Rysunek 9 Przykładowa wizualizacja metody łokciowej



Źródło: *K-means Cluster Analysis*, UC Business Analytics R Programming Guide, 2017, https://uc-r.github.io/kmeans_clustering (dostęp: 25.07.2020)

Miara Silhouette

Metoda ta pozwala na przedstawienie informacji na temat odległości obiektu od wszystkich pozostałych z grupy w odniesieniu do pozostałych obiektów z najbliższego klastra, czyli innymi słowy stopnia dopasowania obiektu do grupy.

Formalny zapis miary Silhouette wyraża wzór (10).

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{10}$$

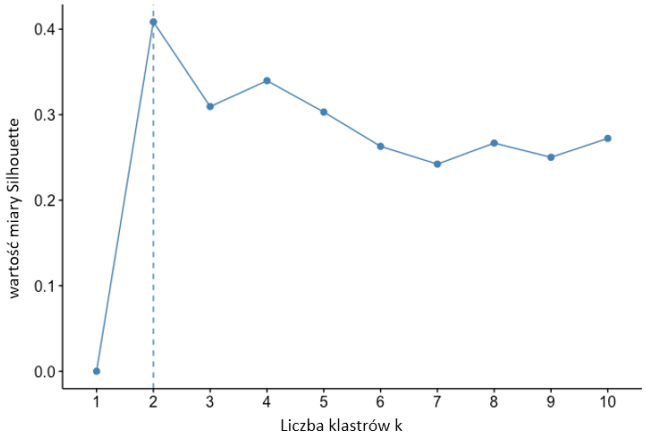
gdzie :

- a(i) - średnia odległość obserwacji x_i od pozostałych z jej grupy,
- b(i) – najmniejsza ze średnich odległości obserwacji od pozostałych grup

Wartości miary s(i) mieszczą się w przedziale od 1 do -1. Wartości bliższe 1 oznaczają lepsze dopasowanie obiektu do grupy, natomiast bliższe -1 oznaczają bliższą odległość od sąsiadującego klastra. Optymalna liczba klastrów zostaje wyłoniona na podstawie maksymalnej wartości indeksu[26]. Analizując przykładowy wykres z *Rysunku 10* wystarczyłoby wziąć pod uwagę 2 klastry, jednak widoczny jest skok wartości miary Silhouette w okolicy grupy 4. Może to oznaczać wystąpienie pewnego zróżnicowania, a więc warto w takiej sytuacji rozważyć większą liczbę klastrów.

[26] C.Patil, I. Baidari, *Estimating the Optimal Number of Clusters k in a Dataset Using Data Depth*, Data Science and Engineering, 2019, s. 134

Rysunek 10 Przykładowa wizualizacja metody wykorzystującej miarę Silhouette



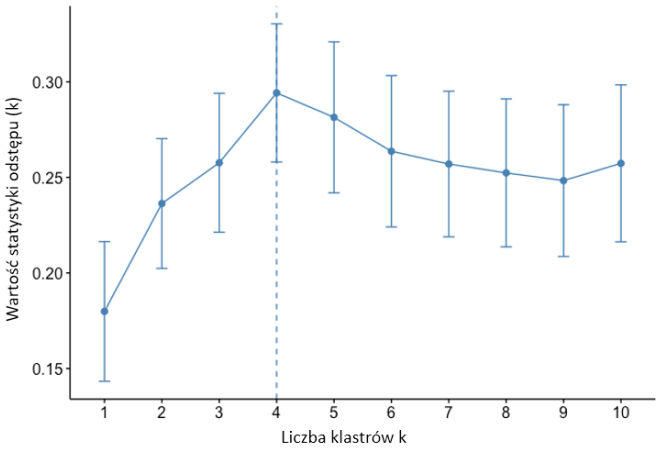
Źródło: *K-means Cluster Analysis*, UC Business Analytics R Programming Guide, 2017, https://uc-r.github.io/kmeans_clustering (dostęp: 25.07.2020)

Statystyka odstępu

Miara ta jest uniwersalna dla większości metod grupowania i opiera się na maksymalizacji tzw. statystyki odstępu wyrażonej wzorem (11). Równanie to uwzględnia średnią W_k^* obliczoną w oparciu o obserwacje, które pochodzą z rozkładu jednostajnego i zostały wygenerowane dla takiego samego przedziału wartości jak w oryginalnym zbiorze danych.

$$Gap_n(\mathrm{k}) = \log(W_k^*) - \log(W_k) \tag{11}$$

Rysunek 11 Przykładowa wizualizacja wartości statystyki odstępu



Źródło: *K-means Cluster Analysis*, UC Business Analytics R Programming Guide, 2017, https://uc-r.github.io/kmeans_clustering (dostęp: 25.07.2020)

Optymalna liczba klastrów jest więc wtedy, gdy różnica między wartością rzeczywistą a losową jest największa[27]. Zgodnie z przykładem na *Rysunku 11* maksymalna wartość współczynnika występuje dla 4 klastrów.

Indeks Calińskiego- Harabasza

Inny sposób na znalezienie końcowej liczby skupień zaproponowany przez Calińskiego i Harabasza zakłada oparcie tego wyboru o wartość pseudo-statystyki *F*, zwanej indeksem Calińskiego- Harabasza, która ma postać zadaną wzorem (12). Dąży się do maksymalizacji indeksu *CH*.

$$CH(K) = \frac{\frac{tr(B(C_k))}{(K-1)}}{\frac{tr(W(C_k))}{(n-k)}}, \tag{12}$$

gdzie $B(C_k)$ to międzyklastrowa, a $W(C_k)$ wewnątrzklastrowa zmienność, które obliczane są jako ślady macierzy[28].

[27] Hastie T., Tibshirani R., Friedman J., *The Elements of Statistical Learning. Data Mining, Inference, and Prediction. Second Edition*, Springer, 2009, s. 519
[28] Górecki T., Krzyśko M., Skorzybut M., Wołyński W., *Systemy uczące się*, Wydawnictwo Naukowo-Techniczne Warszawa, 2008, s.388

3. Przegląd innych algorytmów klasyfikacyjnych

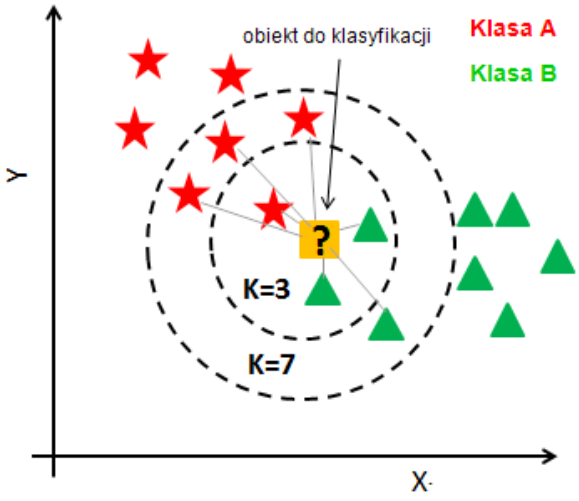
3.1 Metoda K-najbliższych sąsiadów

Metoda najbliższego sąsiada, nazywana dalej KNN, należy do nieparametrycznych sposobów klasyfikacji. Jej zasada działania polega na przydzielaniu danego obiektu do klasy, do której zakwalifikowała się większość z jego *K*-najbliższych sąsiadów. Często metoda ta jest porównywana z grupowaniem za pomocą k-średnich, jednak podstawową różnicą między nimi jest to, iż KNN jest algorytmem uczenia z nadzorem stosowanym do problemów klasyfikacji lub regresji, natomiast *K*-średnich reprezentuje algorytmy bez nadzoru używane do grupowania.

Omówienie elementów algorytmu KNN

Główną część metody KNN stanowi obliczenie odległości pomiędzy obiektami, dzięki którym wyodrębnianych jest K sąsiadów znajdujących się w najmniejszej odległości od badanego elementu. Pojęcie niepodobieństwa obiektów wraz z najczęściej wykorzystywanymi miarami odległości zostały omówione szerzej w przypadku analizy skupień w rozdziale 2.3. Wybór właściwej miary odległości powinien zależeć od kształtu granicy pomiędzy klasami, w szczególności przy punktach leżących w jej pobliżu. *Rysunek 12* przedstawia przykład klasyfikacji obiektu za pomocą KNN.

Rysunek 12 Metoda K-najbliższych sąsiadów



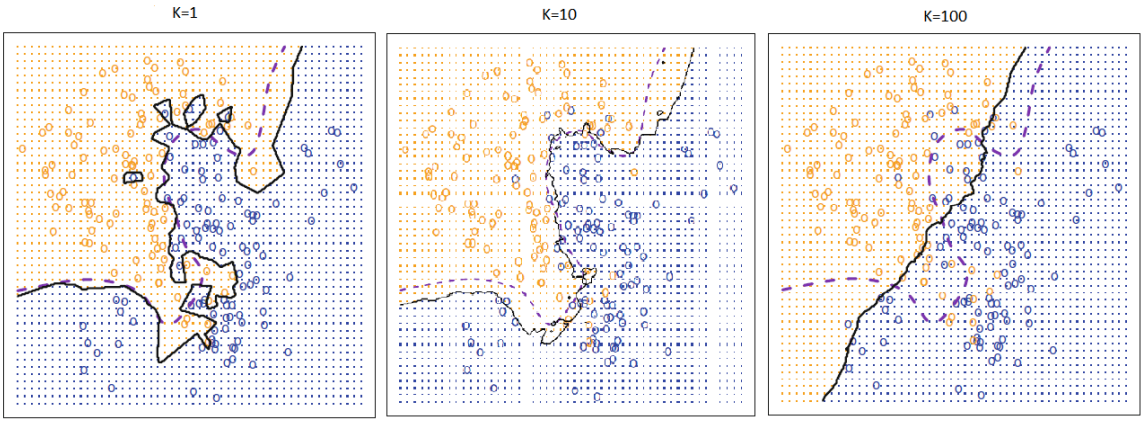
Źródło: Navlani A., *KNN Classification using Scikit-learn*, DataCamp, 2018, https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn (dostęp: 30.07.2020)

Warto mieć na uwadze fakt, że uzasadniony wybór miary niepodobieństwa nie zawsze pozwoli na otrzymanie dobrych wyników. Związane jest to z ryzykiem wystąpienia nieistotnych cech podczas klasyfikacji, które biorą udział jedynie w kształtowaniu otoczeń punktów, a nie granic określających klasy, w wyniku czego wprowadzają zakłócenia w podjęciu decyzji[29].

Metoda *K*-najbliższych sąsiadów daje różne wyniki w zależności od przyjętej liczby sąsiadów. *K* jest podstawowym parametrem, który ma wpływ jakość modelu- w przypadku małych wartości wykaże bardzo dużą zmienność predykcji, natomiast przy zbyt dużych może okazać się niedokładny. Ogólnie mówiąc, liczba sąsiadów powinna być na tyle duża, aby minimalizowała prawdopodobieństwo błędnych klasyfikacji.

Rysunek 13 przedstawia graficznie granice klasyfikacji wykonanej metodą KNN dla różnych liczb najbliższych sąsiadów, kolejno K=1, K=10 i K=100.

Rysunek 13 Granice klasyfikacji dla różnych K w metodzie najbliższych sąsiadów



Źródło: Hastie T., James G., Witten D., Tibshirani R., *An Introduction to Statistical Learning*, Springer, New York, 2013, s. 41

Częstym sposobem na znalezienie złotego środka jest zastosowanie metody *v*-krotnej walidacji krzyżowej, która opiera się na podziale danych na *v*-losowych części. Pierwszy krok tej techniki zakłada przyjęcie wstępnej wartości *K* w celu dokonania predykcji dla *v*-tej grupy. Po tym możliwe jest obliczenie błędu predykcji, który w problemach klasyfikacyjnych jest dokładnością, czyli procentem poprawnych klasyfikacji. Procedura ta powtarzana jest dla wszystkich grup, aby na końcu móc

[29] Koronacki J., Ćwik J., *Statystyczne systemy uczące się. Wydanie drugie*, Akademicka Oficyna Wydawnicza EXIT, 2015, s. 121-124

uśrednić błędy i otrzymać miarę jakości modelu. Algorytm ten wymaga wielu obliczeń, jako że przeprowadzany jest dla różnych *K* w celu wybrania najlepszego[30].

Przed przystąpieniem do analizy za pomocą metody k-najbliższych sąsiadów należy pamiętać o normalizacji wartości cech, która w większości przypadków jest niezbędnym etapem. Zapobiegnie to dominacji pewnych atrybutów oraz przysłanianiu przez nie wpływu pozostałych cech.

Klasyfikator KNN

Metoda najbliższych sąsiadów pozwala na bezpośrednie obliczenie estymatora prawdopodobieństwa a posteriori przynależności obiektu *x* do danej klasy *j* poprzez oszacowanie udziału obserwacji z tej klasy w gronie jej *K* najbliższych sąsiadów, co wyrażone jest wzorem (13):

$$\hat{p}(j|x) = \frac{1}{K}\sum_{i=1}^{n} I(\rho(x, x_i) \leq \rho(x, x^{(K)}))\, I(y_i = j)\ , \quad j = 1, \ldots, J \tag{13}$$

gdzie:

- $x^{(K)}$ jest punktem próby uczącej, który jest K-ty co do odległości od obiektu *x*
- ρ jest pewną miarą niepodobieństwa obiektów (odległością)
- I jest funkcją indykatorową.

Podsumowując klasyfikator w KNN ma postać zapisaną wzorem (14):

$$\hat{d}_{NN}(x) = argmax\, \hat{p}(j|x), \quad \mathrm{j} = 1, \ldots, \mathrm{J} \tag{14}$$

Metoda najbliższych sąsiadów charakteryzuje się dużą efektywnością w przypadku, gdy liczba obserwacji jest duża.

[30] *K-najbliższych sąsiadów*, StatSoft, https://www.statsoft.pl/textbook/stathome_stat.html?https%3A%2F%2Fwww.statsoft.pl%2Ftextbook%2Fstknn.html (dostęp: 30.07.2020)

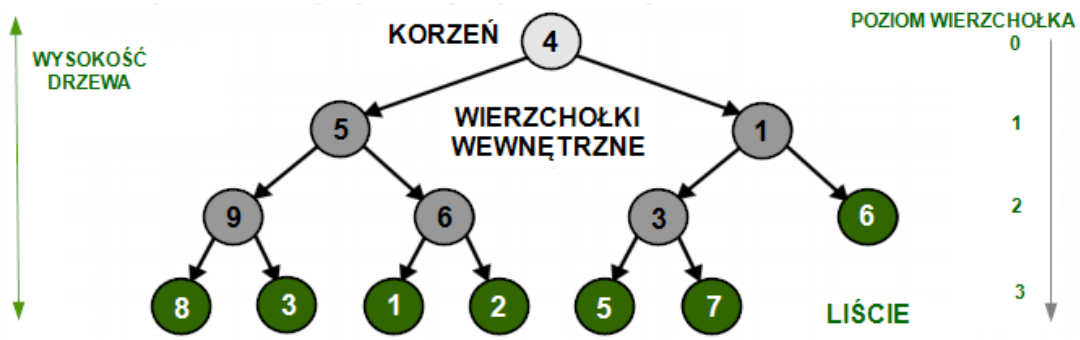
3.2 Drzewa klasyfikacyjne i las losowy

Drzewa decyzyjne to metoda używana zarówno do rozwiązywania problemów regresji, jak i klasyfikacji w zależności od typu zmiennej objaśnianej. Technika ta dąży do ustalenia przynależności obiektów do odpowiednich klas na podstawie analizy zmiennych objaśniających.

Drzewa klasyfikacyjne to nieskierowane lub skierowane grafy, na które składają się następujące elementy umieszczone na *Rysunku 14*:

- początkowy wierzchołek drzewa (korzeń),
- węzły wyjściowe, przejściowe oraz końcowe (wierzchołki),
- krawędzie łączące wierzchołki,
- węzły końcowe (liście),

Rysunek 14 Elementy drzewa klasyfikacyjnego



Źródło: Horzyk A, *Drzewa i struktury drzewiaste*, Wydział EAIIB AGH, 2018, http://home.agh.edu.pl/~horzyk/lectures/wdi/WDI-Drzewa.pdf

Wierzchołki w drzewach prezentowane są w sposób warstwowy za pomocą różnych poziomów. Każdy z nich jest równy drodze od wierzchołka do korzenia. Maksymalny poziom drzewa wyrażony jest poprzez długość najdłuższej drogi prowadzącej od korzenia do poszczególnych liści. W przypadku krawędzi zawsze istnieje tylko jedna ścieżka między dwoma dowolnymi wierzchołkami. Ponadto nie istnieje taki ciąg krawędzi, który łączyłby dany wierzchołek z samym sobą, przez co graf określany jest jako acykliczny[31].

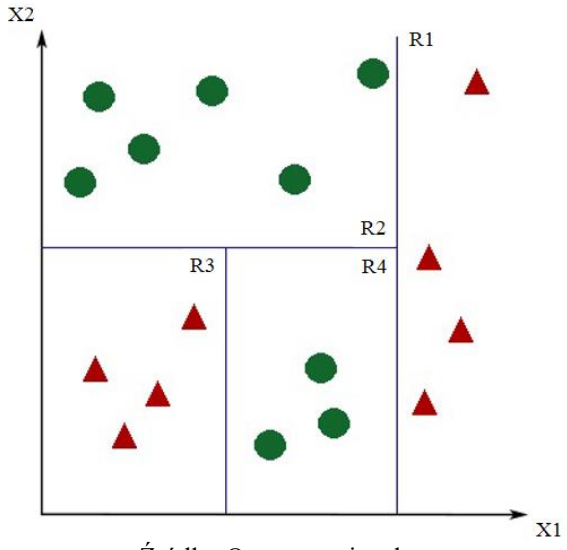
[31] Koronacki J., Ćwik J., *Statystyczne systemy uczące się. Wydanie drugie*, Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2015, s 129-131

Zasada działania drzew klasyfikacyjnych

Konstrukcja drzew opiera się w na algorytmie RP (ang. *recursive partitioning*), która w celu wyodrębnienia najbardziej jednorodnych grup stosuje rekurencyjny podział podzbiorów obserwacji. Punktem wejściowym w konstrukcji drzew klasyfikacyjnych jest optymalny podział zbioru wartości $X_1, X_2, \dots, X_p$, nazywanego przestrzenią predykcyjną, na J podzbiorów $R_1, R_2, \dots, R_J$ tak, aby uzyskać jak najmniejsze zróżnicowanie obiektów w węzłach drzewa. Etap ten obrazuje *Rysunek 15*.

Pierwsza cecha oraz wartości, na podstawie których dokonywany jest podział stanowią korzeń drzewa. Dalsza klasyfikacja przebiega zgodnie z zasadą zstępującą. Jest to algorytm zachłanny, ponieważ na każdym etapie budowy drzewa brany jest pod uwagę najlepszy podział w danym momencie.

Rysunek 15 Przykład klasyfikacji



Źródło: Opracowanie własne

Rekurencyjny podział wymaga wyboru predyktora X_j oraz punktu podziału przestrzeni s, co przedstawia wzór (15).

$$R_1(j,s) = \{X | X_j < s\} \text{ oraz } R_2(j,s) = \{X | X_j \geq s\} \tag{15}$$

Podstawą decyzji o wyborze s jest minimalizacja wybranego kryterium podziału, które zostaną omówione w dalszej części. Algorytm podziału jest powtarzany do osiągnięcia przyjętego kryterium zatrzymania, którym może być np. osiągnięcie maksymalnej liczby obserwacji w danym regionie. Po wyodrębnieniu podzbiorów

$R_1, R_2, \ldots, R_J$ następuje prognoza decyzji o przynależności obserwacji do danej klasy, przypisując jej najczęściej występującą klasę spośród obiektów ze zbioru treningowego znajdujących się w tym samym regionie co badana obserwacja[32].

Kryteria podziału

Wspomniane wcześniej kryteria podziału stosowane w przypadku drzew klasyfikacyjnych to między innymi:

- wskaźnik błędu klasyfikacji

Jest to ułamek obserwacji ze zbioru treningowego, które nie zostały przypisane do najczęstszej klasy. Zależność tę opisuje wzór (16), gdzie $\hat{p}_{mK}$ to odsetek obserwacji ze zbioru treningowego w m-tym węźle, należących do K-tej klasy.

$$E = 1 - max_K(\hat{p}_{mK}) \tag{16}$$

- indeks Giniego oraz entropia

Służą do pomiaru całkowitego zróżnicowania w *K*-klasach. Indeks Giniego jest wyrażony wzorem (17), natomiast entropia wzorem (18). Często współczynniki te określane są jako miara czystości węzła.

$$G = \sum_{k=1}^{K} \hat{p}_{mK}(1 - \hat{p}_{mK}) \tag{17}$$

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} log \hat{p}_{mk} \tag{18}$$

W momencie gdy $\hat{p}_{mK}$ jest równy 0 lub 1, wskaźniki te przyjmują minimalną wartość, co oznacza, że w węźle występują obserwacje zakwalifikowane do tej samej klasy. Indeks Giniego oraz entropia są częściej stosowane niż wskaźnik błędu, ponieważ są one bardziej czułe na zmiany w rozkładzie klas[33].

[32] Hastie T., James G., Witten D., Tibshirani R., *An Introduction to Statistical Learning*, Springer, New York, 2013, s 305-307
[33] Hastie T., James G., Witten D., Tibshirani R., *An Introduction to Statistical Learning*, Springer, New York, 2013, s 311-312

Pruning

Zaprezentowany algorytm często prowadzi do powstawania bardzo złożonych drzew, przez co tego typu modele narażone są na ogromne błędy klasyfikacji. Pojawiają się również sytuacje, w których model bardzo dobrze radzi sobie z klasyfikacją na zbiorze uczącym, jednak jest on do niego nadmiernie dopasowany, co ma wpływ na jakość predykcji klasy nowych elementów w miarę zwiększania się liczby liści. Stosowanie jedynie warunku zatrzymania nie zawsze jest efektywnym podejściem. W związku z tym kolejnym krokiem w budowaniu drzew klasyfikacyjnych jest tzw. *pruning*, czyli przycinanie drzewa, które ma za zadanie zapobiec przetrenowaniu modelu.

Idea tej strategii polega na pozbywaniu się dolnych gałęzi drzewa, dopóki model nie uzyska najwyższej zdolności klasyfikacyjnej. Algorytm ten zakłada utworzenie maksymalnego drzewa T_0 z zamiarem przycinania w celu uzyskania poddrzew o minimalnym błędzie testowym. Metoda przycinania nazywana jest także metodą kosztu złożoności (ang. *cost complexity pruning*), co oznacza, że opiera się na generowaniu sekwencji drzew indeksowanych przy użyciu parametru kar α, który kontroluje równowagę między dopasowaniem do zbioru uczącego a złożonością poddrzewa. Każdemu α odpowiada poddrzewo T zawarte w T_0 określone równaniem (19).

$$\sum_{m=1}^{|T|} \sum_{i:xi \in Rm} (y_i - \hat{y}_{Rm})^2 + \alpha|T|, \tag{19}$$

W tym wzorze *Rm* oznacza podzbiór przedziału prognozy, który odpowiada m-temu węzłowi, natomiast $\hat{y}_{Rm}$ jest prognozą dla tego podzbioru. $|T|$ stanowi liczbę końcowych węzłów drzewa T.

Ostatecznie ze zbioru poddrzew wybierane jest to, które posiada najmniejszy odsetek błędów predykcji na zbiorze testowym. Jednak nie zawsze dostępna jest próba testowa, zatem decyzja jest zazwyczaj wynikiem kroswalidacji[34].

[34] Hastie T., James G., Witten D., Tibshirani R., *An Introduction to Statistical Learning*, Springer, New York, 2013, s 311-312

Istota lasów losowych

Pomimo łatwości w interpretacji oraz dużej intuicyjności w analizie drzew klasyfikacyjnych, wyniki prognoz uzyskanych tą metodą nie zawsze są najdokładniejsze. Ponadto pojedyncze drzewa są bardzo wrażliwe na wszelkie zmiany w danych skutkując pewnymi odchyleniami w ostatecznym wyniku. Nie należy także pominąć faktu iż nieodłącznymi cechami zbiorów danych są braki oraz niewyjaśnione zmienności, które również mogą mieć wpływ na proces uczenia. W związku z tym, w celu polepszenia zdolności klasyfikacyjnych modelu często zamiast pojedynczych drzew wykorzystuje się bardziej złożone metody oparte na rodzinach klasyfikatorów, do których należą między innymi bagging, boosting oraz lasy losowe. Ostatni algorytm zostanie omówiony nieco szerzej.

Procedura lasów losowych (ang. *random forest*) jako uogólnienie drzew decyzyjnych jest zaliczana do algorytmów agregujących i jest swego rodzaju rozszerzeniem baggingu. Ideą tej metody jest zredukowanie korelacji pomiędzy pojedynczymi jednostkami tworzących rodzinę klasyfikatorów bez niepotrzebnego zwiększania wariancji. Z tego względu procedura RF ma przewagę nad baggingiem, ponieważ w każdym podziale uwzględnia tylko część atrybutów.

Algorytm ten stosuje procedurę bootstrapową do wygenerowania wielu zbiorów uczących z populacji n niezależnych obserwacji. Etap ten opiera się na ustaleniu prawdopodobieństwa równego $\frac{1}{n}$, które jest podstawą pobierania obserwacji do nowego zbioru treningowego. Procedura powtarzana jest wielokrotnie i skutkuje powstaniem wielu niezależnych zbiorów używanych przy trenowaniu klasyfikatorów. Należy mieć na uwadze, że niektóre obserwacje mogą zostać wykorzystane wielokrotnie w zbiorach uczących, a niektóre będą całkowicie pominięte.

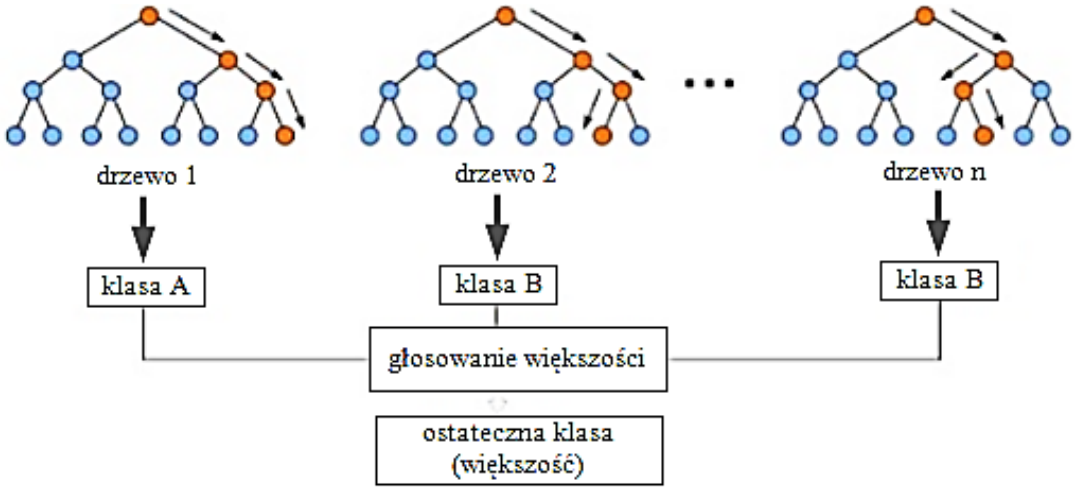
Przy użyciu otrzymanej próby losowej konstruowane są drzewa klasyfikacyjne T_b. W tym celu na początku następuje wylosowanie m spośród p atrybutów. Przyjmuje się, że m jest równe pierwiastkowi kwadratowemu liczby wszystkich predyktorów $m = \sqrt{p}$. Następnie konieczne jest zastosowanie dla nich wybranego kryterium podziału, względem którego nastąpi podział węzła[35].

Klasyfikacja danego wektora obserwacji za pomocą lasu losowego przebiega przez wszystkie drzewa wchodzące w skład lasu, prowadząc do wygenerowania zbioru

[35] Hastie T., James G., Witten D., Tibshirani R., *An Introduction to Statistical Learning*, Springer, New York, 2013, s 316-320

składającego się z B drzew, określanych jako $\{T_b\}_1^B$. Ostateczny wynik określany jest w oparciu o tzw. głosowanie większościowe, polegające na zweryfikowaniu klas w poszczególnych drzewach, do których przypisano daną obserwację oraz przydzieleniu jej do klasy, do której zakwalifikowała ją większość drzew[36]. Algorytm ten w uproszczeniu przedstawia *Rysunek 16.*

Rysunek 16 Algorytm lasu losowego



Źródło: Koehrsen W., *Random Forest Simple Explanation*, Medium, 2017, https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d (dostęp: 25.07.2020)

W lasach losowych weryfikacja błędu testowego przeprowadzana jest przy użyciu obserwacji, które nie zostały wykorzystane podczas procesu i stanowią około $\frac{1}{3}$ zbioru. Określane są jako *out-of-bag* i oznaczają średni błąd dla każdego elementu przy klasyfikacji na podstawie drzew, w których budowie obserwacja ta nie brała udziału. Błąd OOB wykorzystywany jest w ocenie istotności i siły predykcji danego *m*-tego atrybutu poprzez porównanie go z takim samym ułamkiem wyznaczonym dla przekształconej w skutek losowej permutacji próby[37].

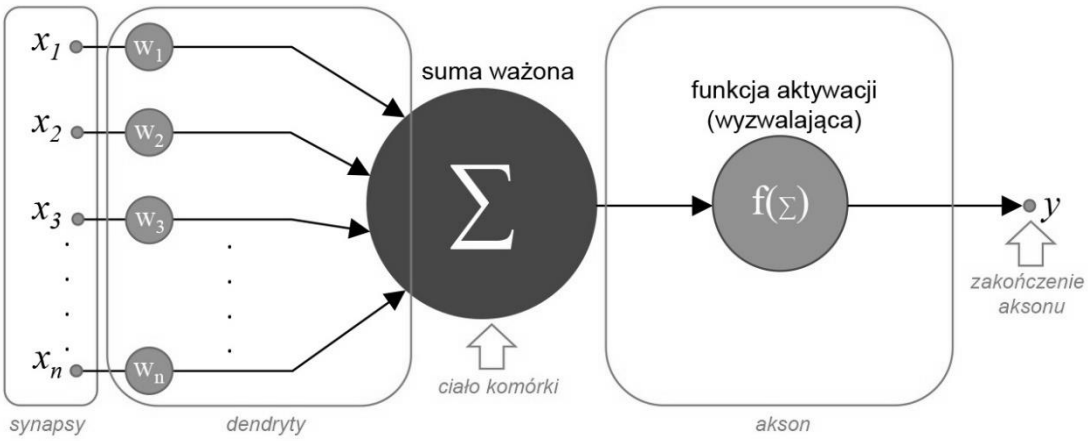
[36] Hastie T., Tibshirani R., Friedman J., *The Elements of Statistical Learning. Data Mining, Inference, and Prediction. Second Edition*, Springer, 2009, s. 606-608
[37] Koronacki J., Ćwik J., *Statystyczne systemy uczące się. Wydanie drugie*, Akademicka Oficyna Wydawnicza EXIT, 2015, s. 164

3.3 Sieci neuronowe

Przy omówieniu kolejnej metody z dziedziny sztucznej inteligencji, jaką są sieci neuronowe, warto zacząć od zdefiniowania pojęcia sztucznego neuronu. Jego pierwowzorem był biologiczny neuron, czyli komórka zdolna do przetwarzania i przekazywania informacji w postaci impulsów nerwowych.

Do podstawowych elementów sztucznego neuronu należą *n*-wejść neuronu wraz z wagami oraz jeden sygnał wyjściowy, co zostało zestawione na *Rysunku 17* z analogią do neuronu biologicznego.

Rysunek 17 Sztuczny neuron wraz z analogiami do neuronu naturalnego



Źródło: Rogalewski P**.,** *AI dla każdego. Część 3*, Czasopismo Zabezpieczenia, 2019, https://www.zabezpieczenia.com.pl/nowe-technologie/ai-dla-kazdego-czesc-3 (dostęp: 03.09.2020)

Sygnały wejściowe z przypisanymi wagami trafiają do bloku, gdzie odbywa się wyliczenie sumy ważonej sygnałów z poszczególnych wejść. Im większa waga przy sygnale wejścia, tym większe ma on znaczenie. Następnie wartość ta jest przekazywana do funkcji aktywacji, a jej wynik do wyjścia *y*. Neuron posiada dodatkowe wejście będące obciążeniem neuronu- bias, które wpływa na wartość funkcji aktywacji [38].

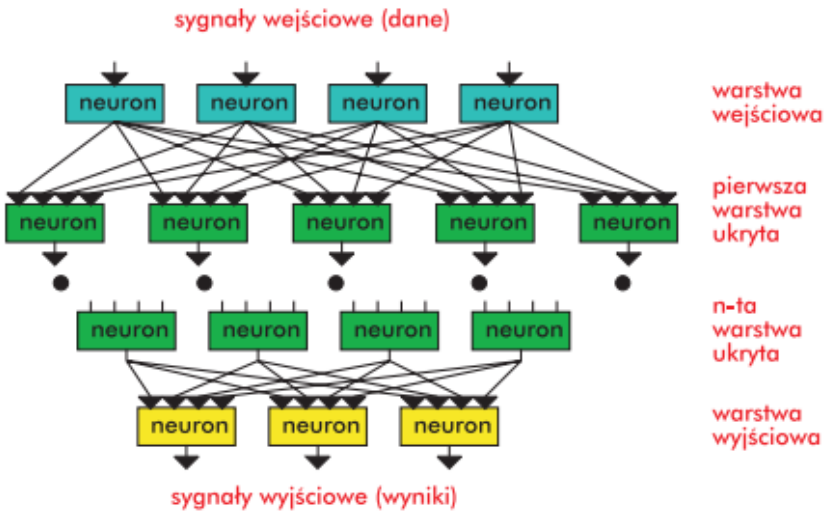
[38] Rogalewski P**.,** *AI dla każdego. Część 3*, Czasopismo Zabezpieczenia, 2019, https://www.zabezpieczenia.com.pl/nowe-technologie/ai-dla-kazdego-czesc-3 (dostęp: 03.09.2020)

Zasada działania wielowarstwowych sieci neuronowych

Najprostszym typem sieci neuronowych jest perceptron, który składa się z jednego lub wielu neuronów typu McCullocha-Pittsa i jest wykorzystywany do uczenia klasyfikatorów binarnych. Jego działanie polega na klasyfikacji danych wejściowych poprzez odpowiednią modyfikację wag i połączeń między warstwami neuronów, tak aby ostatecznie wynik reprezentował oczekiwane wartości. Tego typu podejście ma jednak ograniczenia jeśli chodzi o bardziej złożone modele, dlatego większym zainteresowaniem cieszą się wielowarstwowe sieci neuronowe, zwane MLP (ang. *Multilayer Perceptron*). Dodatkowo perceptron jednowarstwowy może być stosowany jedynie na zbiorach, które są liniowo separowane, w odróżnieniu do MLP. Warto podkreślić, że w podstawowym założeniu sieć MLP nie posiada sprzężenia zwrotnego, czyli jest jednokierunkowa. Przeciwieństwem są sieci rekurencyjne.

Wielowarstwowe sieci neuronowe składają się z warstwy wejściowej, warstw ukrytych oraz warstwy wyjściowej (*Rysunek 17*).

Rysunek 17 Schemat budowy wielowarstwowej sieci neuronowej



Źródło: Tadeusiewicz R., Gąciarz T., B.Borowik, B.Leper, Odkrywanie właściwości sieci neuronowych, Polska Akademia Umiejętności, 2007, s.41

Funkcją warstwy wejściowej jest pobieranie i rozsyłanie sygnałów do najbliższej warstwy ukrytej. Neurony zazwyczaj nie przetwarzają otrzymanych sygnałów lecz dokonują ich rozdystrybuowania do kolejnych warstw. Neurony warstwy ukrytej w kolejnym kroku przetwarzają dane wykorzystując wagi, czyli zgromadzone informacje i generują sygnał wyjściowy przekazywany do kolejnej ukrytej warstwy.

Proces powtarza się dla wszystkich warstw ukrytych kończąc na warstwie wyjściowej, która oblicza wartości i przekazuje je na zewnątrz. Neurony warstwy wyjściowej mają możliwość agregowania sygnałów oraz wykorzystania swoich charakterystyk do skonstruowania końcowego rozwiązania przekazywanego na wyjściu. Mimo tego, że sygnałów warstw ukrytych nie da się zaobserwować w żadnym momencie procesu, są one najbardziej istotnym elementem przy rozwiązywaniu danego problemu, ponieważ w tej warstwie znajduje się najwięcej informacji pozyskanych podczas uczenia. Kluczem do rozwiązania problemu jest to, że sieci działają zawsze jako całość i każdy jej element ma wkład w poszukiwaniu wyniku[39].

Funkcja aktywacji

Wspomniana przy okazji omawiania sztucznego neuronu funkcja aktywacji służy do obliczania wartości wyjścia neuronów. W zależności od modelu neuronu może ona przyjmować różne formy: nieliniową, liniową oraz skoku jednostkowego. Argumentem funkcji aktywacji neuronu są sumy iloczynów sygnałów i ich wag[40], co ogólnie można wyrazić wzorem (20).

$$y = f\left(\sum_{i=1}^{n} x_i w_i\right) \tag{20}$$

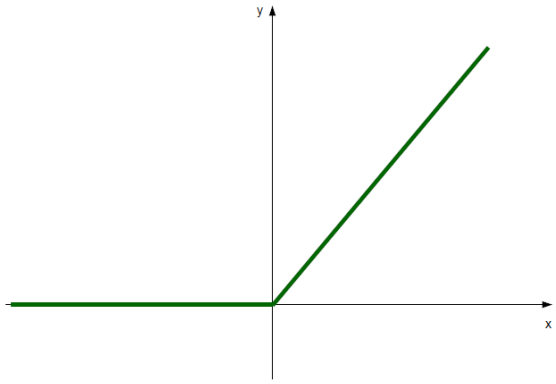
Jedną z najczęściej wykorzystywanych jest jednostronnie obcięta funkcja liniowa będąca połączniem funkcji tożsamości z funkcją progową, nazywana ReLU (ang. *Rectified Linear Unit*) postaci:

$$f(x) = \max(0, x) \tag{21}$$

Dla wartości wejściowych powyżej zera przyjmuje ona wartość wejściową, natomiast w przeciwnym przypadku wartości utrzymują się na poziomie 0, co widać na *Rysunku 18.* Funkcja ta najbardziej przypomina odpowiednik neuronu biologicznego[41].

[39] Tadeusiewicz R., Gąciarz T., B.Borowik, B.Leper, *Odkrywanie właściwości sieci neuronowych*, Polska Akademia Umiejętności, 2007, s.77-79
[40] *Sztuczne sieci neuronowe. Rozpoznawanie wzorców - podejście praktyczne*, http://www.ai.c-labtech.net/sn/pod_prakt.html (dostęp: 03.09.2020)
[41] Horzyk A., *Uczenie głębokie i głębokie sieci neuronowe*, Wydział EAIIB AGH, 2018, http://home.agh.edu.pl/~horzyk/lectures/miw/MIW-DL.pdf (dostęp: 03.09.2020)

Rysunek 18 Funkcja aktywacji ReLU



Źródło: Opracowanie własne

Wybór funkcji aktywacji często zależy od problemu, do którego ma być ona zastosowana. Inne wykorzystywane rodzaje funkcji aktywacji zostały zebrane w *Tabeli 2*.

Tabela 2 Najpopularniejsze funkcje aktywacji

Funkcja aktywacji	**Wzór**
Funkcja liniowa	$f(x) = x$
Jednostronnie obcięta funkcja liniowa dodatnia	$f(x) = \begin{cases} 0, dla\ x \leq 0 \\ x, dla\ x > 0 \end{cases}$
Obcięta symetryczna funkcja liniowa	$f(x) = \begin{cases} -1, dla\ x < -1 \\ x, dla -1 \leq x \leq 1 \\ 1, dla\ x > 1 \end{cases}$
Funkcja progowa unipolarna	$f(x) = \begin{cases} 0, dla\ x < a \\ 1, dla\ x \geq a \end{cases}$
Funkcja progowa bipolarna	$f(x) = \begin{cases} -1, dla\ x < a \\ 1, dla\ x \geq a \end{cases}$
Logarytmiczno- sigmoidalna funkcja unipolarna	$f(x) = \frac{1}{1+e^{-\beta x}}$
Tangens hiperboliczny	$f(x) = \frac{1-e^{-\beta x}}{1+e^{-\beta x}}$
Funkcja Gaussa	$f(x) = ae^{-\frac{(x-b)^2}{2c^2}}$

Źródło: Górecki T., Krzyśko M., Skorzybut M., Wołyński W., *Systemy uczące się*, Wydawnictwo Naukowo-Techniczne Warszawa, 2008, s.206

Mimo swojej prostoty tożsamościowa oraz progowa funkcja aktywacji, które były wykorzystywane w czasach kiedy komputery nie posiadały wystarczającej mocy obliczeniowej coraz częściej są zastępowane funkcjami nieliniowymi, tj. sigmoidalna, tangensa hiperbolicznego i ReLU[42].

Warstwy sieci

Jak zostało wspomniane wcześniej sieć może składać się z wielu warstw ukrytych. Często spełniona jest zasada, że im więcej warstw występuje w sieci, tym bardziej jest ona zdolna do znalezienia głębokich zależności w danych wejściowych. Dodatkowo w modelach istnieje możliwość stosowania warstw różnego typu oraz dostosowania wybranej funkcji aktywacji i liczby neuronów. Parametry będące liczbę warstw i neuronów w nich zawartych wybiera się w sposób gwarantujący jak najmniejszy błąd działania sieci. Sieci, które nie posiadają warstw ukrytych mogą w rezultacie modelować jedynie proste zależności liniowe. Z kolei sieci bardziej rozbudowane mimo tego, że prawie zawsze osiąganą minimalne wartości błędu, mogą oznaczać model przetrenowany, co można zauważyć w momencie, kiedy błąd walidacyjny zaczyna wzrastać przed osiągnięciem satysfakcjonującego poziomu wytrenowania sieci. Najczęstszym sposobem na pozbycie się tego problemu jest usunięcie pewnej liczby neuronów ukrytych lub całych warstw[43].

Istnieje kilka typów warstw ukrytych, z czego omówione zostaną tylko wybrane z nich: warstwa łącząca, gęsta oraz konwolucyjna/splotowa.

[42] Zocca V., Spacagna G., Slater D., Roelants P., *Deep Learning. Uczenie głębokie z językiem Python. Sztuczna inteligencja i sieci neuronowe*, Helion, 2018, s. 56-57
[43] *Sieci neuronowe*, StatSoft, https://www.statsoft.pl/textbook/stathome_stat.html?https%3A%2F%2Fwww.statsoft.pl%2Ftextbook%2Fstneunet.html (dostęp: 03.09.2020)

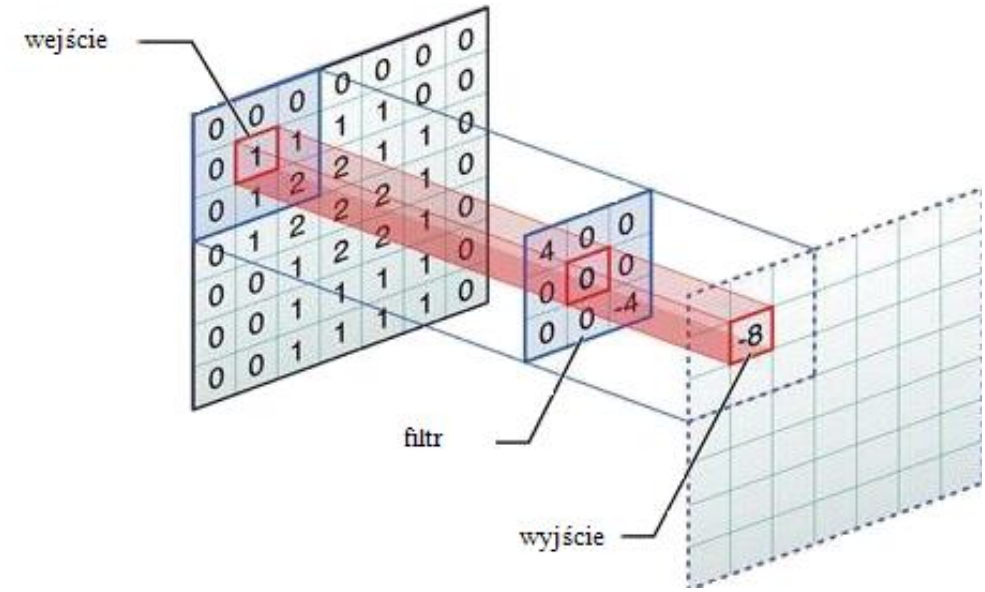
Warstwa splotowa

Warstwa splotowa, czasami określana jako konwolucyjna, służy do wykrywania wzorców w danych wejściowych. W tym celu korzysta z funkcji *h(t),* powstałej w wyniku splotu dwóch funkcji *f(t)* i *g(t),* co wyraża wzór:

$$h(t) = (f * g)(t) = \int_0^t f(x)g(t-x)dx \tag{22}$$

Warstwa ta ma za zadanie filtrowanie sygnału wejściowego f(x) poprzez zastosowanie na nim filtru *g(x)* badającego zależności pomiędzy sąsiednimi elementami. Najczęstsze zastosowanie znajduje ona w przypadku operacji na obrazach 2D, gdzie należy przesuwać filtr w postaci macierzy z wartościami po kolejnych pikselach obrazu. Następne wartości macierzy wyjściowej są obliczane poprzez wyznaczenie sumy pomnożonych przez filtr wartości w danej części obrazu. W ten sposób powstaje nowy obraz utworzony w oparciu o sąsiedztwo pikseli, który umożliwia wyodrębnienie cech[44]. Schemat tego procesu przedstawiony został na *Rysunku 19*.

Rysunek 19 Schemat działania splotowych sieci neuronowych



Źródło: Verma R., *Convolutional Neural Network Basics, 2018* https://rohanverma.net/blog/2018/10/14/convolutional-neural-network-basics/ (dostęp: 03.09.2020)

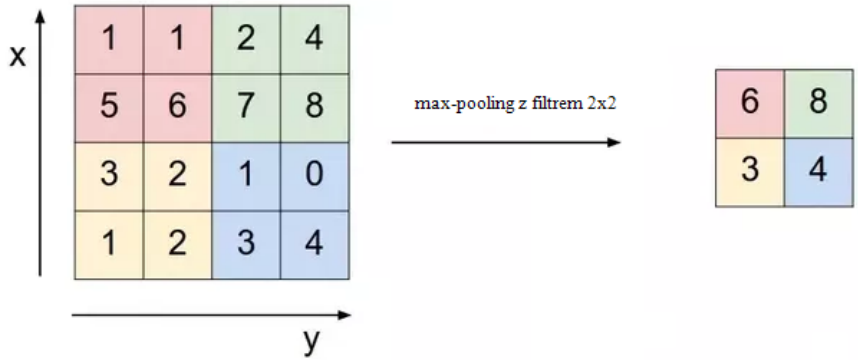
[44] Saha S., *A comprehensive guide to convolutional neural networks*, Medium, 2018 https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 (dostęp 03.09.2020)

W przypadku danych tabelarycznych 1D filtr nakładany jest kolejno na wiersze danych, w zależności od zadeklarowanej wysokości. Tego typu jednowymiarowe sieci splotowe wykorzystywane są do przetwarzania sygnałów lub przetwarzania języka naturalnego.

Warstwa łącząca

Po warstwie splotowej często występuje warstwa łącząca (ang. *pooling*), która służy do redukcji ilości cech danych wejściowych z zachowaniem najważniejszych informacji, a więc zmniejszenia złożoności obliczeniowej modelu sieci. Odpowiada ona także za zminimalizowanie szumu. Najpopularniejszą techniką jest tzw. *max-pooling*, który polega na podziale danych na okna w zależności od rozmiaru filtra oraz wybraniu z każdego z nich największej wartości, tworząc nowy zbiór danych podawany na wejście do następnej warstwy sieci[45]. Metodę tę w uproszczonej wersji przedstawia *Rysunek 20*.

Rysunek 20 Przykład łączenia danych 2D



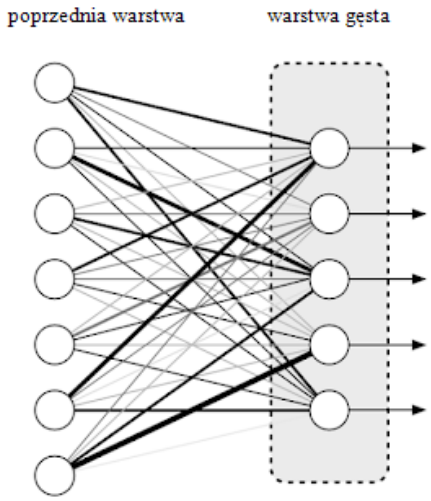
Źródło: Prabhu R., *Understanding of Convolutional Neural Network (CNN) — Deep Learning*, Medium, 2018, https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148 (dostęp: 03.09.2020)

[45] Gibiansky A. *Convolutional Neural Networks*, 2014, https://andrew.gibiansky.com/blog/machine-learning/convolutional-neural-networks/ (dostęp: 03.09.2020)

Warstwa gęsta

Warstwa gęsta (ang. *dense, fully-connected*) jest wykorzystywana zazwyczaj na ostatnich etapach uczenia sieci w celu wsparcia ostatecznej klasyfikacji oraz przeprowadzenia wnioskowania na podstawie cech, które zostały wyodrębnione w poprzednich warstwach. W warstwie gęstej neurony połączone są z neuronami poprzedniej warstwy na zasadzie „każdy z każdym" (jak na *Rysunku 21*), a poszczególne połączenia posiadają swoją wagę stanowiącą informacje pozyskane podczas uczenia sieci.

Rysunek 21 Połączenia neuronów między warstwa gęstą a poprzedzającą



Źródło: *CNN - Fully Connected Layer (FC) - warstwa w pełni połączona,* Blog Ściąga Programisty, 2018, http://sciagaprogramisty.blogspot.com/2018/03/fully-connected-layer-fc-warstwa-w-peni.html (dostęp: 03.09.2020)

Podstawową różnicą między warstwą gęstą, a splotową jest to, że neurony w wejściowych przestrzeniach uczą się cech parametrów globalnych, natomiast w warstwie splotowej uczą się wzorców lokalnych.

Warstwa spłaszczająca

Warstwa spłaszczająca (*ang. flatten*) stosowana jest po warstwie splotowej w celu przekształcenia wielowymiarowej mapy cech w wektor o jednym wymiarze[46].

[46] Mwiti D., *Convolutional Neural Networks: An Intro Tutorial,* Medium, 2018, https://heartbeat.fritz.ai/a-beginners-guide-to-convolutional-neural-networks-cnn-cf26c5ee17ed (dostęp: 16.09.2020)

Warstwa redukcji

Podczas uczenia sieci neuronowych bardzo często dochodzi do tzw. *overfittingu*, czyli nadmiernego dopasowania modelu do zbioru uczącego i tym samym nieumiejętnego klasyfikowania danych nieznanych. Poprzez dodawanie kolejnych warstw sieci oraz zwiększanie liczby filtrów model będzie miał większą możliwość na wychwycenie poszczególnych cech oraz dokładniejszą predykcję klasy. Dlatego też uwzględnia się warstwę redukcji (ang. *dropout*), która w trakcie uczenia w sposób losowy usuwa z sieci pojedyncze neurony zgodnie z podanym prawdopodobieństwem. Zastosowanie takiego podejścia zmusza sieci do uczenia w sposób zgeneralizowany. Warstwa ta jest dodawana zazwyczaj do sieci gęstych, natomiast w przypadku splotowych nie ma ona sensu z uwagi na ideę konwolucji[47].

3.4 Wskaźniki oceny jakości modeli klasyfikacyjnych

Do oceny jakości modeli klasyfikacyjnych służą miary liczbowe oraz graficzne, które umożliwiają porównanie różnych zastosowanych podejść w problemie klasyfikacji.
Podstawowymi oznaczeniami dla liczbowych wskaźników w procesie oceny są:

- *True Positive (TP)* – liczba obserwacji przydzielonych poprawnie do klasy pozytywnej,
- *True Negative (TN)* – liczba obserwacji przydzielonych poprawnie do klasy negatywnej,
- *False Positive (FP)* – liczba obserwacji przydzielonych błędnie do klasy pozytywnej, czyli błąd pierwszego rodzaju,
- *False Negative (FN)* – liczba obserwacji przydzielonych błędnie do klasy negatywnej, czyli błąd drugiego rodzaju.

W przypadku klasyfikacji dwuklasowej sytuacja jest prosta- występują tylko dwie grupy, spośród których jedna jest pozytywna, a druga negatywna. Przy problemach wieloklasowych na klasę negatywną składają się wszystkie klasy poza pozostałą, będącą klasą pozytywną[48].

[47] *Konwolucyjne sieci neuronowe 3: overfitting*, AI Geek Programmer, 2020, https://aigeekprogrammer.com/pl/konwolucyjne-sieci-neuronowe-tutorial-czesc-3/ (dostęp: 16.09.2020)
[48] *Jak ocenić jakość i poprawność modeli klasyfikacyjnych? Część 1 – Wprowadzenie*, Algolytics, https://algolytics.pl/tutorial-jak-ocenic-jakosc-i-poprawnosc-modeli-klasyfikacyjnych-czesc-1-wprowadzenie/ (dostęp: 02.08.2020)

Macierz pomyłek

Macierzą błędów (ang. *confusion matrix*) jest tablica przedstawiona na *Rysunku 22* o wymiarach *n* x *n*, w której wiersze stanowią klasy rzeczywiste, natomiast kolumny przewidywane. Zawarte są w niej liczby obserwacji, które zostały zakwalifikowane do poszczególnych klas, poprzez porównanie stanu faktycznego z klasami wskazanymi przez model[49].

Rysunek 22 Macierz pomyłek dla problemów wielowymiarowych

Źródło: Opracowanie własne

[49] *Jak ocenić jakość i poprawność modeli klasyfikacyjnych? Część 3 – Confusion Matrix*, Algolytics, https://algolytics.pl/jak-ocenic-jakosc-i-poprawnosc-modeli-klasyfikacyjnych-czesc-3-confusion-matrix/ (dostęp: 02.08.2020)

Liczbowe wskaźniki

Miary liczbowe do oceny jakości modelu oparte są na wskaźnikach *TP, TN, FP, FN*. Przy budowie modelu dąży się do minimalizacji liczby błędów, czyli jak najmniejszego *FN* oraz *FP*. Do liczbowych wskaźników należą dokładność, miary zasięgu i pokrycia tj. czułość oraz specyficzność, miary precyzji przewidywania pozytywnego i negatywnego.

Dokładność modelu służy do określenia poprawnych klasyfikacji w stosunku do wszystkich przypisań, a więc definiuje skuteczność zastosowanej reguły decyzyjnej:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{23}$$

Czułość (*True Positive Rate*) określona wzorem (24) informuje w jakim stopniu rzeczywista klasa pozytywna jest zgodna prognozą pozytywną.

$$TPR = \frac{TP}{TP + FN} \tag{24}$$

Analogiczne znaczenie w odniesieniu do klas negatywnych posiada specyficzność (*True Negative Rate*), co oddaje wzór (25).

$$TNR = \frac{TN}{TN + FP} \tag{25}$$

Precyzja przewidywania pozytywnego (*Positive Predictive Value*) określa z kolei w jakim procencie prognoza klasy pozytywnej pokrywa się ze stanem faktycznie pozytywnym[50] i obliczana jest zgodnie ze wzorem (26).

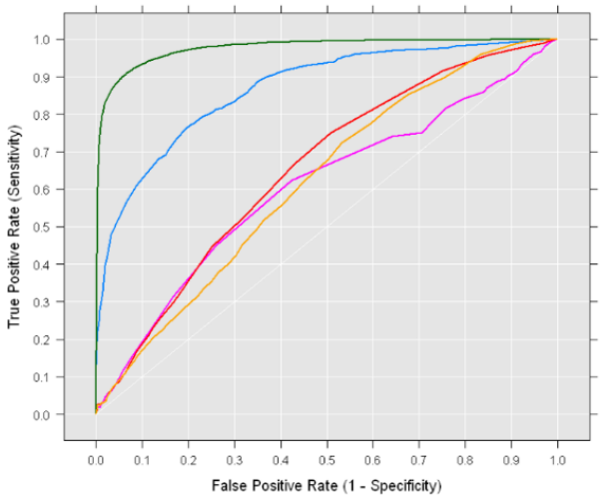
$$PPV = \frac{TP}{TP + FP} \tag{26}$$

[50] *Zasięg (TPR – czułość / TNR – specyficzność) i precyzja (PPV / NPV) – czyli ocena jakości klasyfikacji (część 2),* Mathspace, http://mathspace.pl/matematyka/ocena-jakosci-klasyfikacji-czesc-2/ (dostęp: 02.08.2020)

Krzywa ROC oraz wskaźnik AUC

Efektywność modelu predykcyjnego można zweryfikować graficznie przy użyciu krzywej ROC, która ilustruje zależność czułości i specyficzności (*Rysunek 23*). Wykres ten jest funkcją przyjętego punktu odcięcia. Każda reguła decyzyjna podlega ocenie na podstawie analizy tych dwóch miar. W przypadku równych kosztów błędnych klasyfikacji optymalnym punktem odcięcia jest punkt o współrzędnych (0,1), co oznacza, że wykryte zostały wszystkie obiekty wybranej klasy (czułość= 1) oraz nie ma żadnego obiektu błędnie przypisanego do wyróżnionej klasy (specyficzność= 0). Reasumując, im bardziej wypukły wykres, tym klasyfikator ma lepsze zdolności predykcji.

Rysunek 23 Krzywa ROC



Źródło: Chan C., *What is a ROC Curve and How to Interpret It*, Displayr, 2018, https://www.displayr.com/what-is-a-roc-curve-how-to-interpret-it/ (dostęp: 08.09.2020)

Do interpretacji krzywej ROC najpopularniejszym podejściem jest wyznaczenie pola powierzchni pod wykresem, nazywanego współczynnikiem AUC (ang. *area under curve*). Może być on stosowany jako miara trafności modelu. Przyjmuje wartości od 0 do 1 przy oznaczeniu, że im większa wartość, tym lepszy model. Niewskazane jest obliczanie jedynie współczynnika AUC, ponieważ kształt i przebieg krzywej ROC również pozwala na uzyskanie istotnych informacji na temat wpływu charakteru zmiennej na dany stan[51].

[51] Harańczyk G., *Krzywe ROC, czyli ocena jakości klasyfikatora i poszukiwanie optymalnego punktu odcięcia*, StatSoft, 2010, https://media.statsoft.pl/_old_dnn/downloads/krzywe_roc_czyli_ocena_jakosci.pdf (dostęp: 02.08.2020)

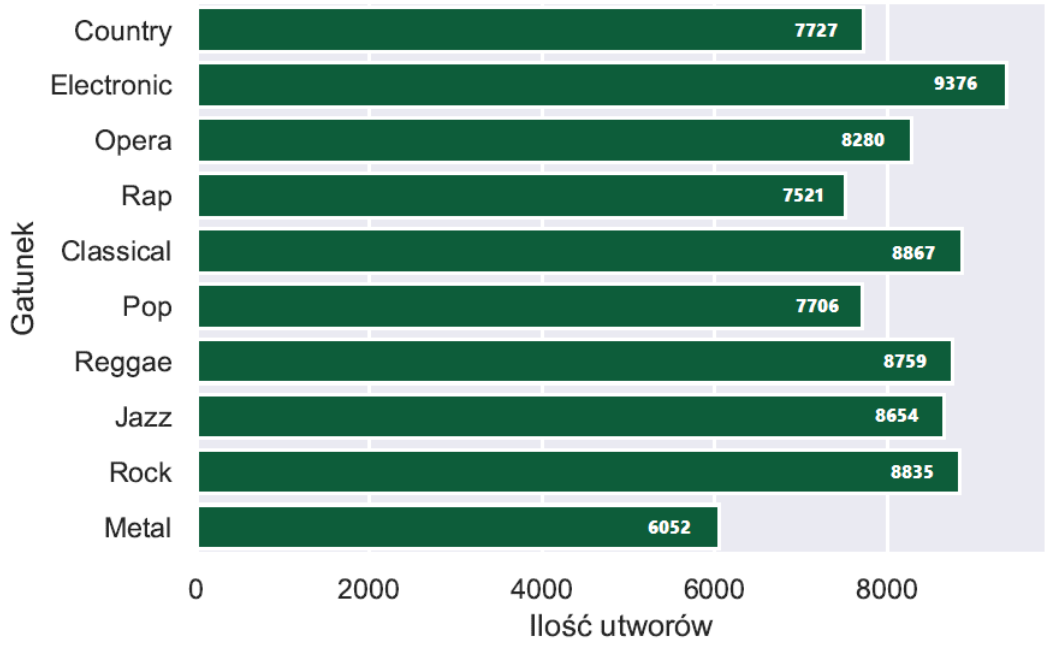
4. Badanie empiryczne

4.1 Przedstawienie danych

Opisane w poprzednim rozdziale metody grupowania i klasyfikacji zostaną zastosowane do określenia gatunków utworów w serwisie Spotify. Na dane wykorzystane w tej części pracy składają się przykładowe piosenki dla każdego z wybranych dziesięciu gatunków muzycznych: elektroniczna, klasyczna, rock, jazz, metal, reggae, rap, pop, operowa, country.

Łącznie zbiór danych zawiera 81777 przykładowych utworów. Na wstępie pominięte zostały zduplikowane utwory, czyli takie, które przypisywane są jednocześnie do kilku gatunków ze względu na różne czynniki. Jest to zjawisko jak najbardziej normalne, jednak obecność powtarzających się wierszy z różnymi kategoriami mogłaby wprowadzić niejasności podczas klasyfikacji i spowodować dużą losowość w przypisywaniu klas. Liczność obserwacji dla poszczególnych gatunków przedstawiona została na *Rysunku 24*.

Rysunek 24 Liczba przykładowych utworów dla poszczególnych gatunków



Utwory muzyczne charakteryzowane są za pomocą 13 cech audio możliwych do pobrania poprzez interfejs webowy Spotify. Zostały one omówione w *Tabeli 3.*

Tabela 3 Cechy audio utworów

Nazwa cechy	**Opis**	**Wartości**
duration_ms	Czas trwania utworu.	Wyrażany w milisekundach.
key	Estymowana tonacja utworu zapisana za pomocą standardowej notacji klas wysokości dźwięku.	'C' →0, 'C#' → 1, 'D' → 2, 'D#' → 3, 'E' → 4, 'F' →5, 'F#' → 6, 'G' → 7, 'G#' → 8, 'A' → 9, 'A#' → 10, 'B' → 11
mode	Skala utworu (durowa/majorowa lub molowa/minorowa), z której pochodzi jego zawartość melodyczna.	Major→ 1 (skala durowa/majorowa) Minor→ 0 (skala molowa/minorowa)
time_signature	Metrum, czyli układ akcentów w obrębie taktu.	0/4→0, 1/4→1, 2/4→2, 3/4→3, 4/4→4, 5/4→5
acousticness	Akustyka utworu.	Przedział od 0,0 do 1,0, gdzie 1 oznacza utwór wykonywany przy użyciu instrumentów akustycznych.
danceability	Stopień taneczności utworu określający czy piosenka nadaje się do tańca w oparciu o tempo, stabilność rytmu, siłę uderzenia i ogólną regularność.	Przedział od 0,0 do 1,0, gdzie 0 określa utwór mało taneczny, a 1 przeciwnie.
energy	Energiczność i intensywność utworu, określona m.in. dzięki szybkości, głośności oraz częstotliwości początkowej.	Przedział od 0,0 do 1,0, gdzie 1 charakteryzuje utwory energiczne i głośne (np. metal), natomiast 0 ścieżki stonowane (np. muzyka klasyczna).
instrumentalness	Poziom występowania wokalu w utworze. Wyrażenia dźwiękonaśladowcze jak „ohhh", „uuuu" traktowane są jako instrumenty.	Przedział od 0,0 do 1,0, gdzie 1 odnosi się do piosenek niezawierających wokalu, natomiast im bliżej 0, tym więcej słów zostało wykrytych w utworze (np. rap). Wartości powyżej 0,5 charakteryzują utwory instrumentalne.

liveness	Obecność publiczności w utworze, najczęściej dotyczy nagrań z koncertów.	Przedział od 0,0 do 1,0. Wyższe wartości, szczególnie od poziomu 0,8, świadczą o tym, że utwór najprawdopodobniej był grany na żywo.
loudness	Głośność utworu uśredniona dla całej ścieżki.	Wyrażana w dB, gdzie typowy zakres to wartości od -60 do 0 dB.
speechiness	Poziom obecności słów w ścieżce.	Przedział od 0,0 do 1,0. Dla nagrań typu audiobook, podcast, poezja wartość ta jest bliska 1. Wartości powyżej 0,66 opisują ścieżki, które prawdopodobnie składają się w całości z mówionych słów. Wartości od 0,33 do 0,66 opisują utwory, które mogą zawierać zarówno muzykę, jak i mowę (np.rap).
valence	Miara określająca nastrój utworu.	Przedział od 0,0 do 1,0. Wartości wysokie dotyczą utworów pozytywnych i radosnych, natomiast niskie odnoszą się do bardziej negatywnych, smutnych lub stonowanych.
tempo	Tempo utworu.	Wyrażane w uderzeniach na minutę BPM.

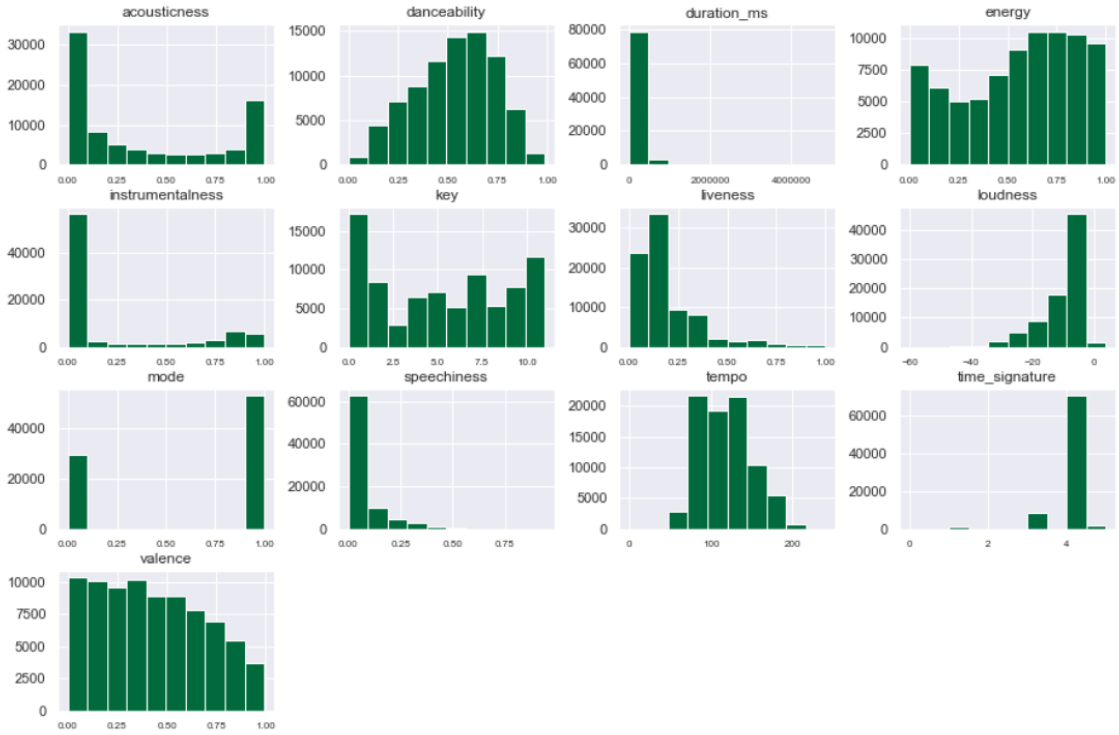
Źródło: *Spotify for developers: Get Audio Features for a Track*, Spotify AB, 2020, https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/ (dostęp: 02.09/2020)

Badanie pozwoli stwierdzić, czy dostępne cechy dźwiękowe ścieżki są wystarczające do jednoznacznego określenia gatunku utworu. Dodatkowo dane zawierają tytuł utworu oraz jego ID, które z punktu widzenia budowy modelu nie są zbyt istotne, jednak zostały zachowane dla przejrzystości.

Rozkład empiryczny cech dźwiękowych

Rozkład cech dla przyjętych danych został zobrazowany na histogramach (*Rysunek 25*). Jak widać w większości utworów występuje wokal, niewielka część zbioru jest całkowicie instrumentalna (*instrumentalness*). Jeśli chodzi poziom słów w utworach, to przykładowe piosenki są raczej połączeniem muzyki i słów, jednak widocznych jest kilka wyjątków (*speechiness*). Dodatkowo przeważa liczba piosenek mniej akustycznych (*acousticness*). Znaczna część przykładów nadaje się do tańca i jest energiczna (*danceability/energy*). Nastrój piosenek jest dość zbalansowany, jednak ilościowo więcej występuje utworów pozytywnych (*valence*). Wyraźną przewagę widać w utworach w skali durowej (*mode*), co wiąże się poniekąd z nastrojem utworów, jako że skala majorowa jest charakterystyczna dla utworów radosnych. Może się to przełożyć na duże powiązanie zmiennych.

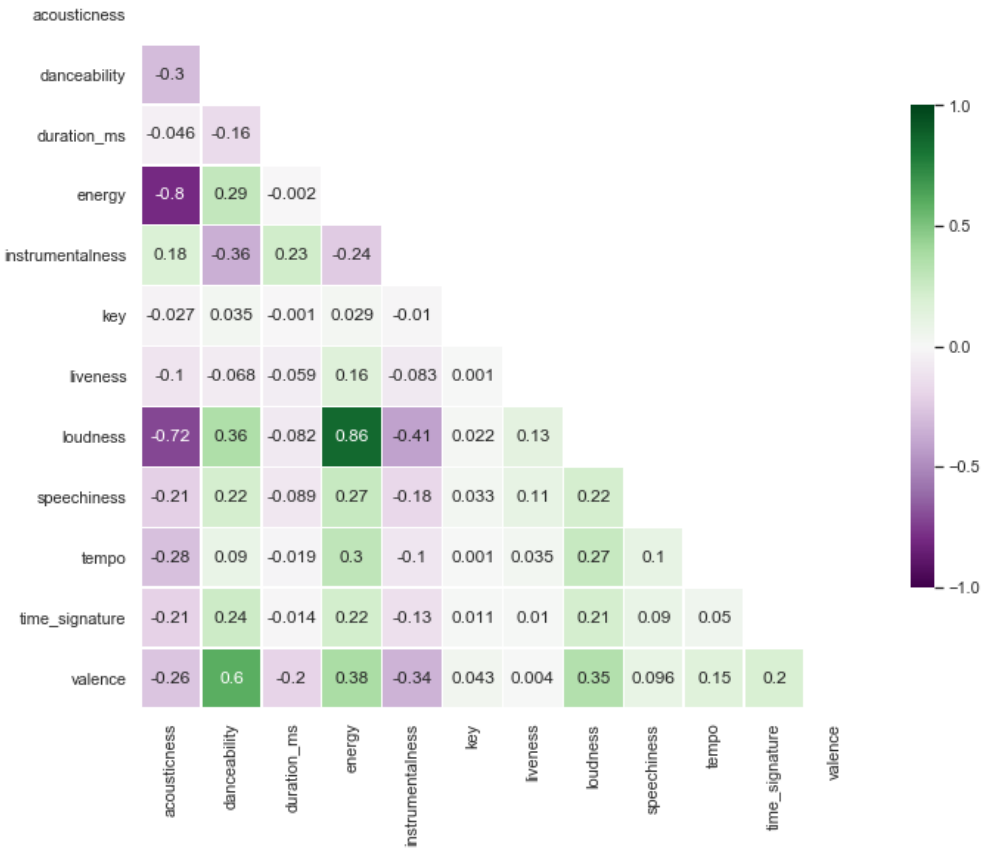
Rysunek 25 Rozkład empiryczny cech audio



Badanie korelacji zmiennych

Przed przystąpieniem do konstrukcji modeli należało zbadać korelację między zmiennymi objaśniającymi, ponieważ pojawienie się bardzo powiązanych ze sobą cech może spowodować współliniowość w późniejszym modelu. Takie zmienne rzadko wprowadzają lepszy poziom objaśnienia zmiennej zależnej, a nawet może to spowodować utratę mocy przewidywania na rzecz innego predyktora. Na *Rysunku 26* przedstawiona została macierz współczynników korelacji badanych cech utworów.

Rysunek 26 Macierz korelacji



Na tej podstawie można stwierdzić wysoką korelację dodatnią między głośnością i energicznością utworu. Jest to poniekąd prawda zgodnie ze specyfiką cechy *energy* opisaną wcześniej, mówiącą o tym, że utwory energiczne są zazwyczaj charakteryzowane na podstawie szybkości oraz głośności. Obie zmienne są dodatkowo odwrotnie skorelowane z cechą dotyczącą akustyczności ścieżki dźwiękowej, co również ma sens ze względu na to, że utwory akustyczne najczęściej kojarzą się z brzmieniem gitary klasycznej, a więc mają bardziej nastrojowy charakter, w przeciwieństwie do muzyki energicznej. Z tego powodu należy mieć na uwadze wysoki poziom zależności zmiennych *energy, loudness, acousticness* i rozważyć ich

wykluczenie w grupowaniu lub budowie modeli klasyfikacyjnych, ponieważ pozostawienie wszystkich może prowadzić do zwiększenia, a nawet podwojenia znaczenia danej zmiennej przy obliczaniu odległości pomiędzy obserwacjami. Decyzję tę należy podejmować jednak ostrożnie, ponieważ usunięcie którejś ze zmiennych może spowodować przykładowo sztuczne zmniejszenie odległości pomiędzy klastrami.

Standaryzacja danych oraz wydzielenie zbioru uczącego i testowego

Przed przystąpieniem do grupowania oraz klasyfikacji koniecznym etapem jest przeskalowanie zmiennych z uwagi na to, że nie wszystkie cechy wyrażone są w takich samych jednostkach oraz zakresie. Poza zmiennymi zawierającymi wartości z przedziału od 0 do 1 w zbiorze występują zmienne w skali porządkowej (*key, time_signature*), nominalnej (*mode*) oraz określone za pomocą innych wartości (*loudness*). Różnice te powodują, że cechy posiadają odmienne rozkłady. Jest to jednoznaczne z tym, że zmienne z większymi odchyleniami standardowymi będą miały również większy wpływ na końcowy wynik.

Standaryzacja polega na transformacji każdej cechy indywidualnie w taki sposób, aby znajdowała się w takiej samej skali z jednakowym rozkładem. Tego typu przekształcenie należy wykonać osobno na zbiorze uczącym i testowym, ponieważ do późniejszej walidacji dokładności modelu wymagane są nowe, nieznane dane. W przypadku odwrotnej kolejności, dane które trafiłyby do zbioru testowego mogłyby mieć duży wpływ na wartość średniej i odchylenia standardowego. Dlatego też w pierwszym kroku należy skalować zbiór uczący, a w kolejnym testowy z użyciem parametrów rozkładu każdej cechy otrzymanych po standaryzacji części treningowej.

Ostatecznie dane, które będą wykorzystane w badaniu mają postać przedstawioną poniżej na *Rysunku 27*.

Rysunek 27 Fragment danych po standaryzacji

acousticness	danceability	duration_ms	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signature	valence
-0.936197	0.339530	0.251372	0.661616	1.745300	0.213102	-0.822827	0.072327	0	-0.641371	-0.886466	0.249362	-0.220169
0.821019	-0.714445	-0.816733	0.310105	-0.613545	0.778859	-0.109380	0.893889	1	-0.497509	1.375454	-1.914383	2.044765
-0.953208	1.192748	-0.461434	0.136090	-0.613877	-1.201292	0.044115	0.276602	1	0.957709	1.685284	0.249362	0.128283
-0.645741	0.680817	0.014691	-0.034445	-0.613697	-1.484171	-0.287435	0.252434	1	-0.629198	0.512389	0.249362	0.594147
-0.934353	-0.077041	-0.587385	0.578089	-0.613860	-1.484171	-0.628195	0.625453	1	-0.552840	1.201338	0.249362	-0.098968

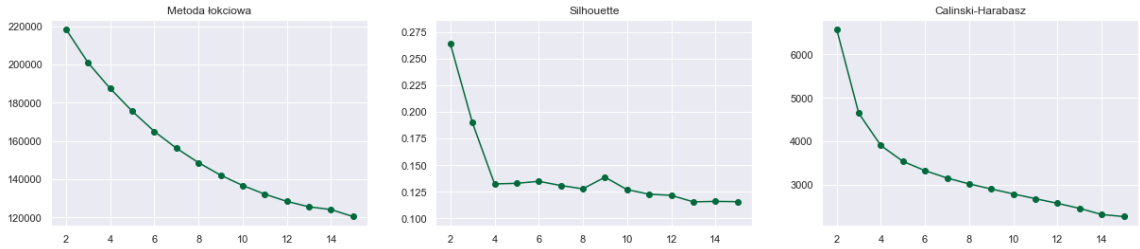
Podział na zbiór uczący i testowy został dokonany w sposób losowy z zachowaniem balansu w liczebności utworów należących do poszczególnych gatunków, gdzie część testowa stanowi 30% wszystkich danych.

4.2 Grupowanie

W części dotyczącej analizy skupień rozważany jest jedynie zbiór testowy z pominięciem zmiennej *energy* z uwagi na wysoką korelację z *loudness* i *acousticness* przekraczającą wartość 0,8. Zdecydowano się na pozostawienie głośności i akustyczności, z uwagi na to, że pominięcie tych zmiennych mogłoby potencjalnie wpłynąć na zbyt duże zmniejszenie odległości pomiędzy skupieniami.

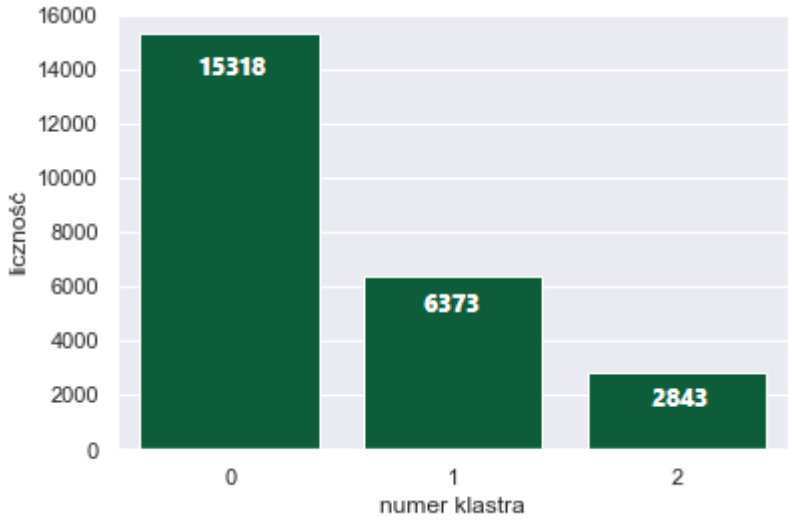
Grupowanie przeprowadzone zostało za pomocą dwóch algorytmów: *K*-średnich oraz metody hierarchicznej Warda. W przypadku pierwszej techniki klasteryzacji przy wyborze liczby *K* wykorzystano 3 miary: metodę łokciową, Silhouette oraz Calińskiego-Harabasza, których wyniki zobrazowane zostały kolejno na wykresach na *Rysunku 28*.

Rysunek 28 Wykresy współczynników decydujących o liczbie klastrów



Pierwszy wykres od prawej nie posiada wyraźnego „łokcia" początkującego stabilizację wartości łącznej wariancji, zatem trudno na jego podstawie zdecydować, w którym momencie należałoby zakończyć dalszy podział zbioru. Dokonując subiektywnej oceny wykresu, można przyjąć, że przy 7 klastrach miara ta wykazuje coraz mniejsze zmiany. W przypadku miary Silhouette oraz Calińskiego-Harabasza wartością maksymalną jest 2, co sugeruje rozważenie jedynie dwóch grup utworów, jednak z uwagi na liczność zbioru i dość wysokie wartości tych współczynników dla *K*=3, ostatecznie metodę *K*-średnich zastosowano do zidentyfikowania trzech grup. Liczebność populacji, która trafiła do poszczególnego skupienia została przedstawiona na *Rysunku 29*. Można zauważyć, że znaczna większość zbioru została zaklasyfikowana do grupy nr 0.

Rysunek 29 Liczebność populacji w trzech klastrach wydzielonych przez algorytm *K*-średnich



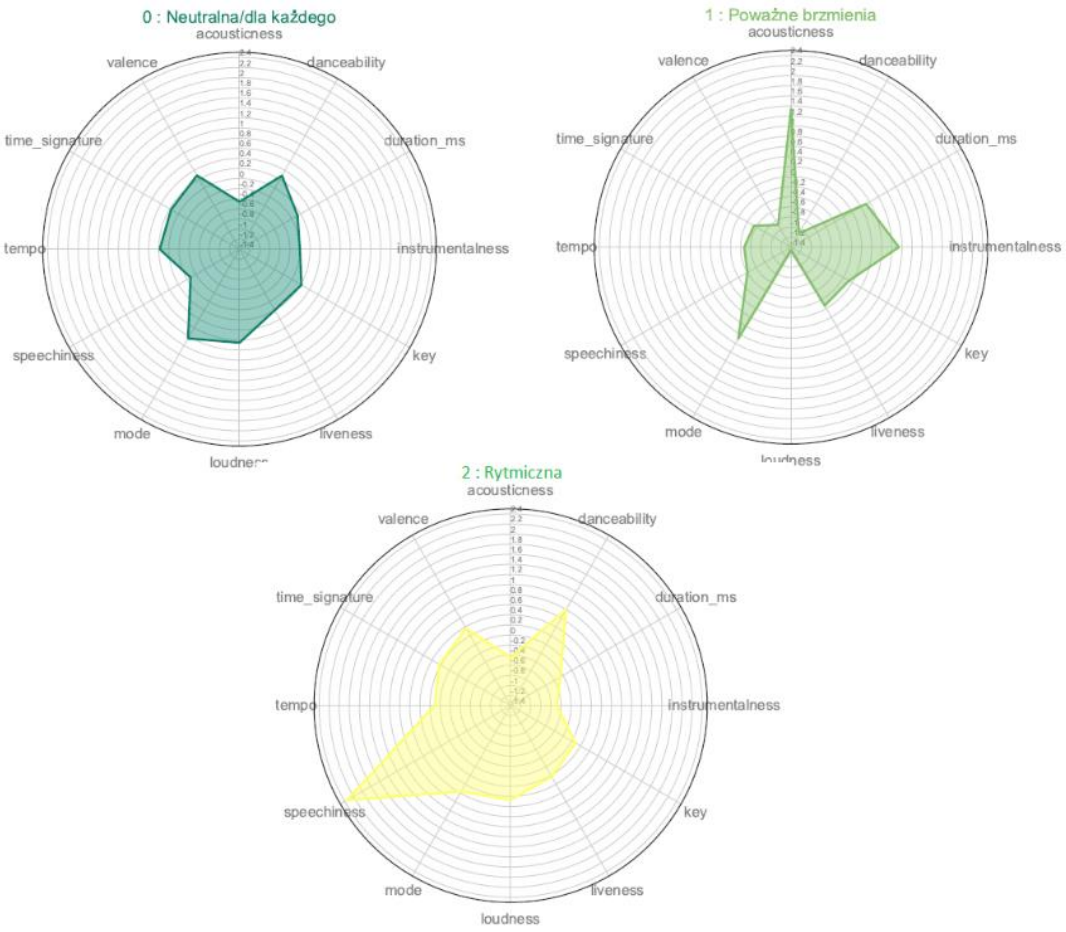
Wyznaczone grupy utworów mogą stanowić trzy odrębne listy odtwarzania w serwisie Spotify, charakteryzując się różnymi cechami, które użytkownik wybierze w zależności od upodobań muzycznych czy nastroju. W celu zbudowania profilu każdej playlisty należy przyjrzeć się bliżej centroidom, czyli średnim wartościom cech audio utworów wchodzących w skład danego klastra, które zostały umieszczone w *Tabeli 4*.

Tabela 4 Centroidy klastrów dla metody K-średnich

numer klastra	acousticness	danceability	duration_ms	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signature	valence
0	-0,46	0,28	-0,07	-0,20	0,02	0,00	0,45	0,64	-0,28	0,18	0,17	0,29
1	1,35	-1,08	0,32	0,74	-0,07	-0,06	-1,34	0,69	-0,40	-0,47	-0,55	-0,89
2	-0,44	0,80	-0,26	-0,48	0,08	0,23	0,47	0,56	2,35	0,11	0,23	0,37

Dodatkowo centroidy zostały zobrazowane na wykresach radialnych na *Rysunku 30* w celu łatwiejszej interpretacji.

Rysunek 30 Profile klastrów uzyskanych metodą *K*-średnich: playlista neutralna, poważna oraz rytmiczna



Playlista 0

Można powiedzieć, że grupa nr 0 zawiera wiele utworów w skali majorowej, na co wskazuje wysoka wartość *mode*. Najczęściej skala ta reprezentuje piosenki o zabarwieniu pozytywnym. Podobne znaczenie posiada cecha *valence*, jednak ze względu na jej wartość bezpieczniej jest określić grupę 0 jako neutralną pod względem nastroju. W przypadku oceny *danceability*, część utworów zapewne będzie nadawała się do tańca ze względu na dość rytmiczne brzmienie i umiarkowane tempo. Piosenki z tej grupy charakteryzują się również zbalansowanym poziomem muzyki i wokalu (*instrumentalness, speechiness*). Zależności te widoczne są na pierwszym wykresie z *Rysunku 30*. Wnioski na podstawie *Tabeli 4* ze środkami klastrów pokrywają się z wizualizacją, podkreślając zbalansowane wartości dla każdej z cech. Można więc pokusić się o określenie tej grupy muzyką neutralną, która przypadnie do gustu

większości ludzi, a więc potencjalnie mogą do niej przynależeć utwory z pogranicza rocka, popu czy elektroniki.

Playlista 1

Grupa ta przedstawiona na drugim wykresie *Rysunku 30* charakteryzuje się wysokim poziomem instrumentalności (*instrumentalness*) i długim czasem trwania utworów (*duration_ms*), co jest często spotykane w muzyce o brzmieniu klasycznym. Dodatkowo wartości dla *acousticness* wskazują na występowanie instrumentów akustycznych w ścieżkach. Wartości cechy liveness wskazują na to, że część utworów może pochodzić z nagrań koncertowych. Zdecydowanie ta lista odtwarzania nie będzie pasowała do tańca (danceability), jako że piosenki te nie cechują się rytmicznością i należałoby je raczej zaliczyć do grona utworów o poważnym nastroju (*valence*). Na podstawie takiego opisu, grupą tą mogłyby zainteresować się osoby, które słuchają muzyki klasycznej lub nastrojowej.

Playlista 2

Spośród wszystkich cech ostatniej grupy utworów należy wyróżnić bardzo wysoki poziom występowania słów (*speechiness*). Piosenkom z taką zawartością tekstu towarzyszy rytmiczna i szybka muzyka (*danceability, tempo*). Muzyka ta nie jest agresywna w wyrazie, przeważa zdecydowanie pozytywny wydźwięk (*valence, mode*). W porównaniu do grup poprzednich, cecha *instrumentalness* posiada najniższe wartości, co świadczy o ogromnej liczbie słów występujących w utworze. Jest to cecha charakterystyczna głównie dla utworów rapowanych, zatem może to być głównym czynnikiem przy wyborze niniejszej listy odtwarzania.

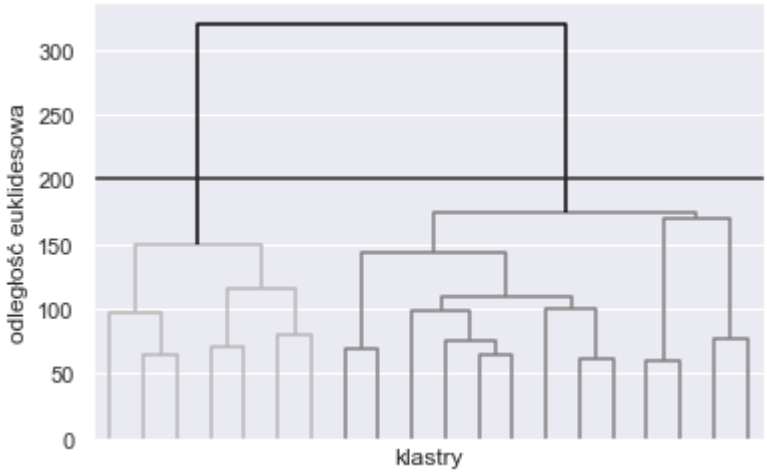
Oprócz stworzenia ogólnych profili poszczególnych list odtwarzania, sprawdzono liczności przeważających gatunków, które reprezentują daną grupę utworów, co przedstawiono w *Tabeli 5*. Jak zauważono na podstawie centroidów, grupę 0 tworzą piosenki reprezentujące głównie takie gatunki jak rock, muzyka elektroniczna, country, reggae oraz pop, a więc taka lista byłaby idealna dla osób ze zróżnicowanym gustem muzycznym. Klaster 1 to z kolei skupisko utworów operowych, klasycznych oraz jazzowych. Ostatnia grupa to zdecydowanie najbardziej rytmiczne utwory charakterystyczne dla brzmień rapowych.

Tabela 5 Rozkład poszczególnych gatunków w klastrach uzyskanych metodą K-średnich

0		1		2	
Rock	2366	Opera	2486	Rap	1090
Electronic	2294	Classical	2356	Reggae	571
Country	2113	Jazz	722	Electronic	403
Reggae	2067	Metal	257	Pop	323
Pop	1862	Electronic	187	Jazz	201
Jazz	1622	Rock	160	Metal	72
Metal	1549	Country	96	Country	70
Rap	1124	Pop	90	Rock	64
Classical	243	Rap	11	Opera	29
Opera	78	Reggae	8	Classical	20

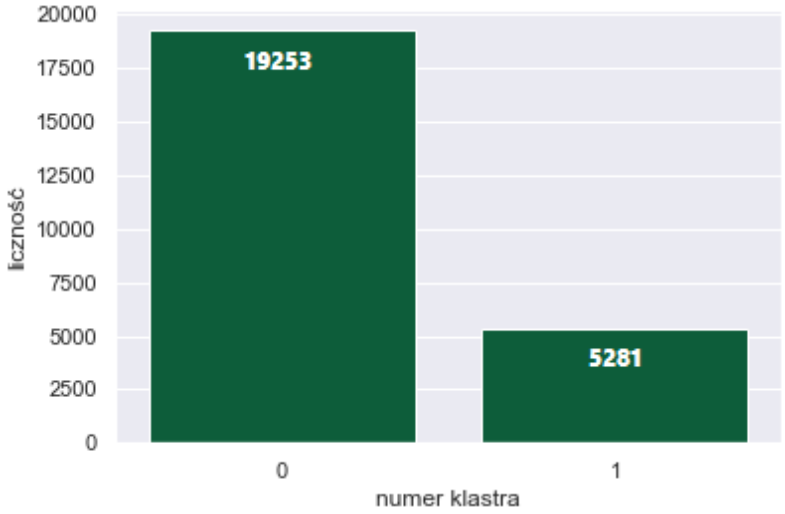
Procesu grupowania dokonano również z użyciem metody Warda z grupy algorytmów hierarchicznych. Decyzję o liczbie klastrów podjęto na podstawie dendrogramu z *Rysunku 31*. W oparciu o największą odległość euklidesową odpowiednią liczbą grup jest 2. Jak widać zastosowanie trzech klastrów tak jak w poprzedniej metodzie nie miałoby sensu ze względu na niewielką odległość pomiędzy skupieniami. Na dendrogramie widać również wzrost odległości przy 5 lub 6 grupach, a więc przedyskutowany zostanie również taki podział.

Rysunek 31 Dendrogram dla metody hierarchicznej



W przypadku jedynie dwóch klastrów znaczną przewagę w liczebności posiada grupa 0, zawierając ponad 19 tysięcy utworów. Przy takim rozkładzie jak na *Rysunku 32* można domyślać się, że tylko częściowo grupy te pokryją się z rezultatami metody *K*-średnich.

Rysunek 32 Liczebność populacji w trzech klastrach wydzielonych przez algorytm *K*-średnich



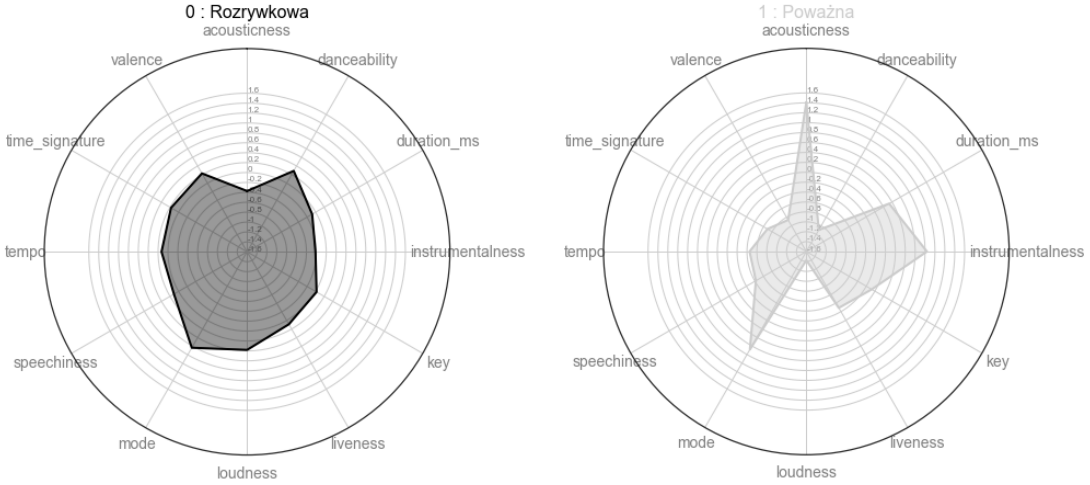
Zgodnie z *Tabelą 6* zawierającą wartości dla centroidów dwóch otrzymanych grup można powiedzieć, że są one dość odległe od siebie ze względu na skrajne wartości dla poszczególnych cech.

Tabela 6 Centroidy klastrów dla metody hierarchicznej

numer klastra	acousticness	danceability	duration_ms	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signature	valence
0	-0,37	0,29	-0,08	-0,22	0,02	0,09	0,38	0,64	0,11	0,13	0,18	0,23
1	1,42	-1,10	0,33	0,84	-0,07	-0,29	-1,42	0,68	-0,42	-0,45	-0,69	-0,86

Pierwszy klaster w tym przypadku charakteryzuje muzykę nadającą się do tańca i śpiewu z uwagi na wysoki współczynnik taneczności oraz niski instrumentalności, która świadczy o dużej ilości występujących słów w utworze. W odróżnieniu od tego skupienia, grupa druga stanowi muzykę bardziej cichą, wolniejszą oraz poważniejszą, w której zostało wykrytych bardzo dużo instrumentów, wpływając na poziom akustyczności. Podobnie jak poprzednio, wartości dla dwóch klastrów zostały zaprezentowane na *Rysunku 33*.

Rysunek 33 Profile klastrów uzyskanych metodą hierarchiczną: playlista rozrywkowa oraz poważna



Przy hierarchicznym grupowaniu można stwierdzić, że playlista neutralna oraz rytmiczna z poprzedniej metody poniekąd zostały połączone w jedną grupę, którą można nazwać ogólnie muzyką rozrywkową. Drugą grupę stanowią brzmienia muzyki klasycznej i operowej, co bardzo dobrze oddaje *Tabela 7* zawierająca rozdystrybuowanie poszczególnych gatunków w klastrach wydzielonych przy użyciu metody Warda.

Tabela 7 Rozkład poszczególnych gatunków w klastrach uzyskanych metodą hierarchiczną dla 2 grup

0		1	
Electronic	2756	Classical	2191
Reggae	2632	Opera	1991
Rock	2498	Jazz	611
Rap	2213	Electronic	128
Pop	2207	Rock	92
Country	2197	Metal	92
Jazz	1934	Country	82
Metal	1786	Pop	68
Opera	602	Reggae	14
Classical	428	Rap	12

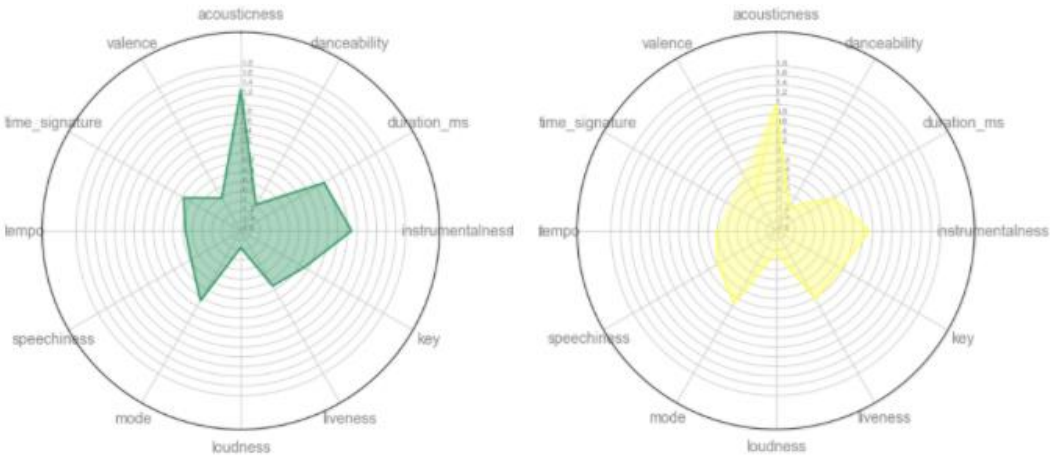
Stosując drugą opcję grupowania zbioru utworów na większą liczbę skupień porównany został podział na 5 oraz 6 grup. Jak można się domyślać na podstawie rozdystrybuowania utworów do poszczególnych list odtwarzania zawartego w *Tabeli 8*, grupa nr 5 jest mało liczna, a utwory do niej należące przy podziale na mniejszą liczbę klastrów trafiają go grup nr 0 oraz 1. Ponadto *Rysunek 34* uwydatnia podobieństwo grupy 1 oraz 5 ze względu na bliskie umiejscowienie centroidów, które sprawia,

że większość utworów z najmniej licznego skupienia przy podziale na mniejszą liczbę kategorii trafia do grupy nr 1. W związku z tym, analizie poddane zostanie grupowanie utworów do 5 list odtwarzania.

Tabela 8 Rozkład utworów przy podziale hierarchicznym na 5 i 6 grup

Numer grupy	5 grup	6 grup
0	6269	6205
1	6205	5975
2	1398	1398
3	2649	2649
4	8013	8013
5	-	294

Rysunek 34 Profile grup 1 oraz 5 uzyskanych metodą hierarchiczną



W efekcie ponowne grupowanie wskazało następujące listy odtwarzania:

- grupa 0 – duży współczynnik wskazujący na występowanie instrumentów przy jednoczesnym dużym poziomie instrumentalności (muzyka klasyczna, operowa)
- grupa 1 – muzyka głośniejsza, taneczna, w dużej mierze instrumentalna (muzyka elektroniczna, jazz),
- grupa 2 – utwory, w których wykryte zostały oklaski i obecność publiczności, a więc muzyka na żywo (muzyka operowa),
- grupa 3 – muzyka rytmiczno-taneczna z dużą liczbą słów (rap, reggae),
- grupa 4 – utwory głośne oraz bardzo rytmiczne, ale ze zbalansowanym występowaniem muzyki i słów (country, rock, pop, reggae).

Centroidy dla wszystkich grup przedstawia *Rysunek 35*, natomiast liczności poszczególnych gatunków, które pojawiają się w danej grupie zebrane zostały w *Tabeli 9*.

Rysunek 35 Profile klastrów uzyskanych metodą hierarchiczną: playlista poważna, taneczna, koncertowa, rytmiczna, rockowa

Tabela 9 Rozkład poszczególnych gatunków w klastrach uzyskanych metodą hierarchiczną dla 5 grup

0		1		2		3		4	
Classical	2273	Electronic	1510	Opera	318	Rap	1010	Country	1711
Opera	2175	Jazz	929	Electronic	204	Reggae	561	Rock	1441
Jazz	802	Metal	864	Reggae	173	Electronic	332	Pop	1160
Rock	211	Reggae	751	Rock	125	Pop	327	Reggae	1119
Metal	207	Rock	738	Jazz	110	Jazz	190	Metal	659
Electronic	191	Pop	575	Country	107	Rock	75	Electronic	647
Country	180	Rap	480	Rap	106	Country	61	Rap	592
Pop	151	Country	220	Classical	104	Metal	59	Jazz	514
Reggae	42	Classical	120	Metal	89	Opera	18	Classical	106
Rap	37	Opera	18	Pop	62	Classical	16	Opera	64

Przedstawione metody grupowania pozwalają na klasyfikację utworów w sposób bardzo ogólny, zwracając szczególną uwagę na wyodrębnienie grup podobnych utworów, co dla wielu użytkowników serwisu Spotify może okazać się bardzo pomocne w przypadku, gdy nie szukają konkretnego gatunku muzyki zależy im jedynie na dostosowaniu muzyki do wybranej okazji.

4.3 Budowa modeli klasyfikacyjnych

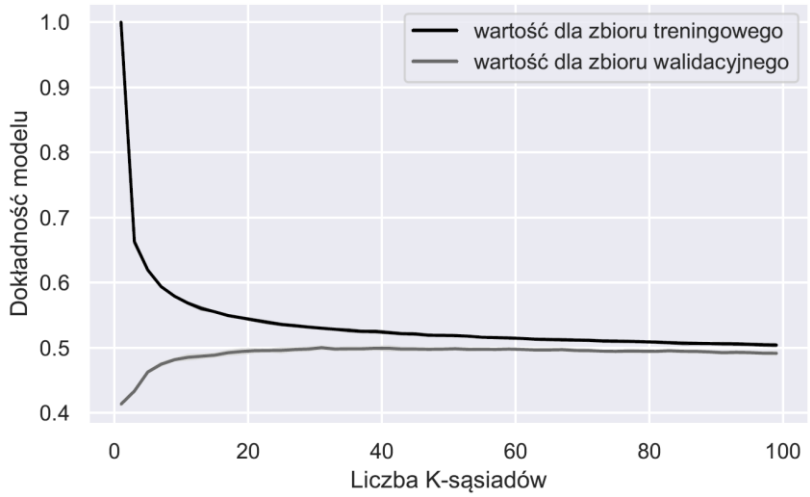
Klasyfikacji utworów dostępnych w analizowanym zbiorze dokonano przy użyciu metod *K*-najbliższych sąsiadów, drzew decyzyjnych, lasów losowych oraz sieci neuronowych. W doborze parametrów modeli pomocne były różne metody oparte głównie na walidacji krzyżowej. Przebieg budowy modeli klasyfikacyjnych oraz ich ulepszeń zostanie przedstawiony kolejno dla każdego algorytmu w kolejnym etapie pracy. Do podstawowej oceny jakości modeli posłuży miara dokładności (*accuracy*), natomiast ostateczne wersje zostaną omówione nieco dokładniej za pomocą macierzy pomyłek oraz miar obliczonych na jej podstawie. Większość modeli zbudowanych zostanie przy użyciu biblioteki *scikit-learn* z języka *Python,* która przeznaczona jest do uczenia maszynowego. W przypadku głębokich i splotowych sieci neuronowych zastosowane zostaną algorytmy z modułu *TensorFlow*.

KNN

Do wytrenowania modelu metodą *K*-najbliższych sąsiadów, został wykorzystany zbiór uczący, który podobnie jak przy grupowaniu nie zawiera skorelowanej z tanecznością i akustycznością zmiennej *energy*.

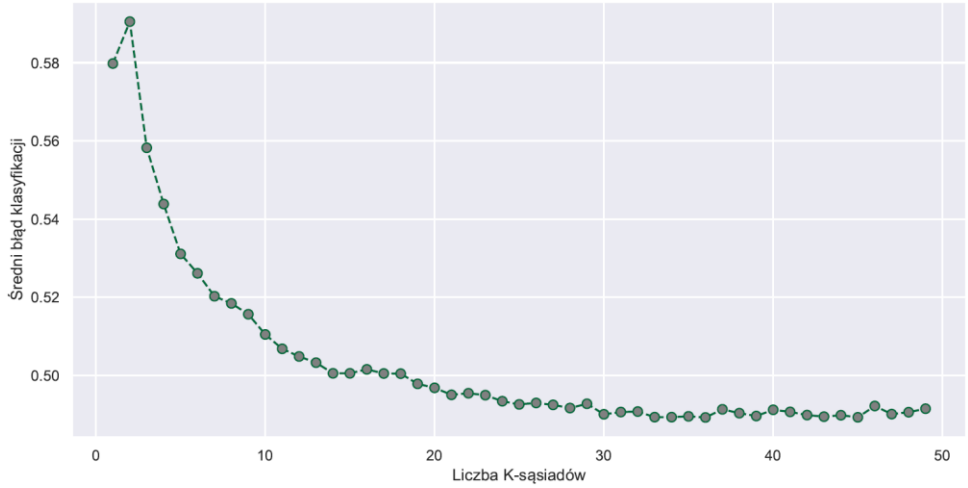
Przed budową właściwego modelu przeprowadzona została walidacja dokładności modelu w zależności od liczby sąsiadów uwzględnianych w modelu, co zostało przedstawione na *Rysunku 36*. Dokładność modelu dla zbioru treningowego wyraźnie spada już dla niespełna 5 sąsiadów. W przypadku zbioru walidacyjnego, który został wydzielony w tej metodzie przy użyciu 3-krotnej kroswalidacji, model zachowuje się praktycznie tak samo, z wyjątkiem znikomych skoków, m.in. dla K=36. Tak czy inaczej, nie należy się spodziewać bardzo trafnych predykcji, jako że dokładność tego modelu to około 50%.

Rysunek 36 Krzywa walidacyjna modelu dla metody KNN



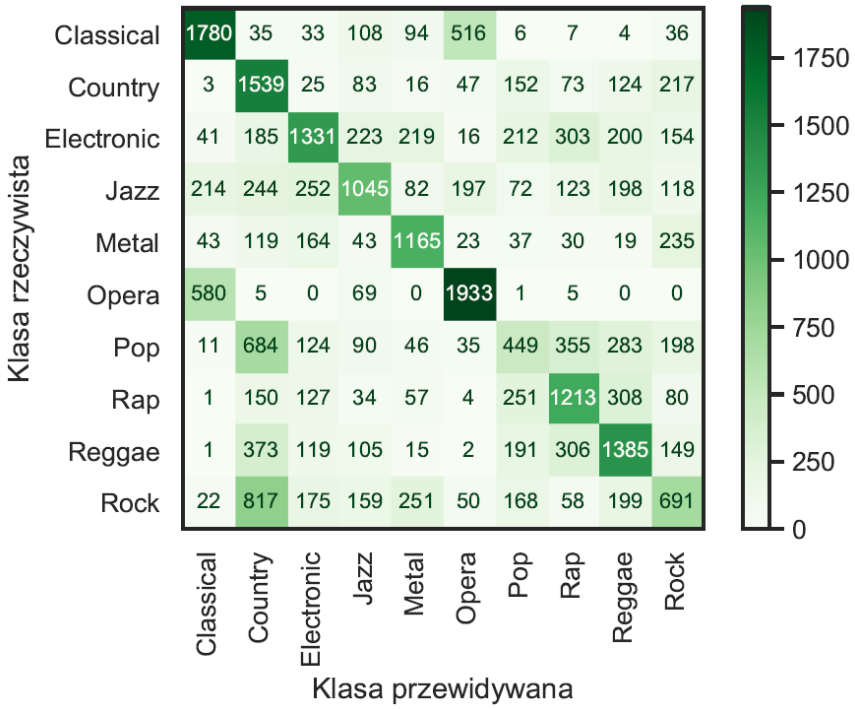
Ostatecznego określenia liczby sąsiadów dokonano na podstawie błędów klasyfikacji uzyskanych dla 50 klasyfikatorów uwzględniających kolejno od 1 do 50 sąsiadów, co przedstawiono na *Rysunku 37*. Zgodnie z zasadą najmniejszej wartości błędu, zbudowany został model KNN dla *K*= 36 sąsiadów.

Rysunek 37 Wartości błędu klasyfikacji w zależności od liczby sąsiadów



Wynik klasyfikacji metodą *K*-najbliższych sąsiadów dla danych testowych został przedstawiony za pomocą macierzy pomyłek na *Rysunku 38*. Pierwszym wnioskiem wynikającym z tej ilustracji jest to, że najbardziej trafne predykcje gatunków w oparciu o cechy audio okazały się być dla muzyki operowej oraz klasycznej. Świadczy to o tym, iż wybrane cechy utworów całkiem dobrze oddają charakter takich utworów. Muzyka operowa jako jedyna ze wszystkich nie została błędnie przypisana do rocka, reggae, metalu oraz muzyki elektronicznej.

Rysunek 38 Macierz pomyłek dla modelu KNN



Przy dokładniejszym spojrzeniu na *Tabelę 10*, która zawiera liczbę wszystkich utworów ze zbioru testowego poddawanych klasyfikacji wraz z precyzją przewidywania pozytywnego i czułością modelu dla poszczególnych gatunków, model wykazuje najwyższą zdolność do wykrywania gatunku operowego. Wychwycił on 1933 utwory operowe z 2593 wszystkich przykładów z tej klasy, osiągając czułość 75%. Nieco gorszą zdolność pokrycia stanu rzeczywistego z wykrytą klasą model ten posiada dla muzyki klasycznej, country oraz metalu, o czym świadczy wartość powyżej 60%. Najczęściej klasyfikator myli się w przypadku utworów pop oraz rock, co oznacza, że może on mieć problemy ze znalezieniem piosenek z tych gatunków, przypisując je niepoprawnie do innych klas. Analizując miarę precyzji, można powiedzieć z kolei, że zdolność prognozy gatunku utworu, która pokrywa się ze stanem faktycznym procentowo jest najwyższa również w przypadku opery oraz muzyki klasycznej. Jest to oznaką tego, że klasyfikator w tych przypadkach najrzadziej przypisuje inny gatunek niż ten właściwy. Jednak przykładowo przy muzyce country, dla której model ten wykazywał się dużą czułością, występuje tendencja do błędnego przypisywania utworów do tej klasy, mimo tego, że w rzeczywistości pochodzą one z innego gatunku.

Tabela 10 Miary dokładności, precyzji oraz czułości modelu KNN

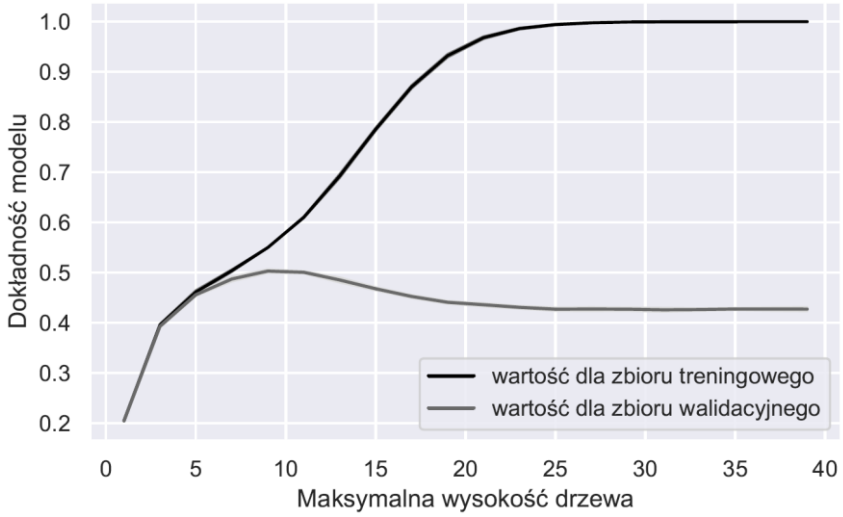
	Precyzja	**Czułość**	**Liczba utworów w danej klasie rzeczywistej**
Classical	0,66	0,68	2619
Country	0,37	0,68	2279
Electronic	0,57	0,46	2884
Jazz	0,53	0,41	2545
Metal	0,60	0,62	1878
Opera	0,68	0,75	2593
Pop	0,29	0,20	2275
Rap	0,49	0,55	2225
Reggae	0,51	0,52	2646
Rock	0,37	0,27	2590
Dokładność modelu	0,511		24534

Ogólna dokładność modelu KNN oszacowana została na zaledwie 51% przy nieznanych danych, a więc gatunki przypisywane są w wielu przypadkach bardzo losowo. Klasyfikator ten nie posiadał również zbyt trafnych predykcji na zbiorze uczącym, osiągając wartość 53%.

Drzewo klasyfikacyjne

Podobnie jak w poprzedniej metodzie, zbiór na podstawie którego uczono model nie zawierał zmiennej *energy*. Na początku dokonana została analiza dokładności modelu dla zbioru uczącego oraz walidacyjnego w zależności od przyjętej maksymalnej wysokości drzewa. Na *Rysunku 39* dokładnie widać, że klasyfikator ten osiąga trafność powyżej 90% w przypadku części treningowej przy wysokości 25, jednak z uwagi na bardzo niską dokładność predykcji gatunku w zbiorze walidacyjnym w tym samym przedziale należy stwierdzić przetrenowanie modelu. Mimo tego, że bardzo dobrze radzi sobie na zbiorze treningowym, nie sprawdzi się on na nowych danych. Dlatego w tym przypadku najlepiej przyjąć 9 jako wysokość drzewa.

Rysunek 39 Krzywa walidacyjna dla drzewa klasyfikacyjnego

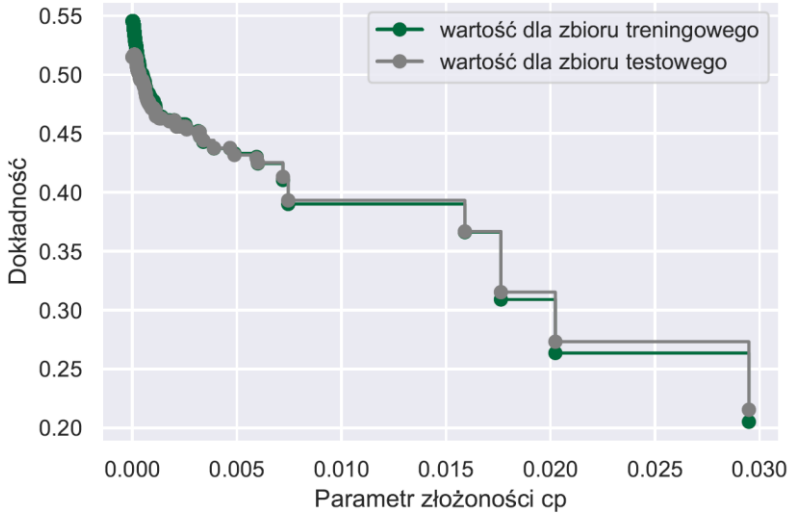


Pierwszy model posiada więc następujące wartości parametrów:

- minimalna ilość próbek w węźle potrzebnych do dokonania podziału = 4
- minimalna ilość obserwacji w liściu = 1
- maksymalna wysokość drzewa = 9

W budowie drzew klasyfikacyjnych często wykorzystywany jest również parametr złożoności *cp*, który służy do kontrolowania rozmiaru drzewa. W przypadku gdy koszt dodania kolejnej zmiennej do drzewa przekroczy wartość tego parametru, dalsze rozrastanie drzewa jest zatrzymywane. Sprawdzony został wpływ tego parametru na dokładność modelu, jednak zgodnie z *Rysunkiem 40* model zachowuje się podobnie przy zbiorze testowym oraz treningowym. Wartość *cp* powinna więc zostać przyjęta na poziomie bliskim 0. Ostatecznie model oparty na drzewie klasyfikacyjnym posiada wcześniej zdefiniowane parametry oraz parametr złożoności równy 0,0000025, osiągając 54,5% trafności na zbiorze uczącym oraz 51,5% na zbiorze testowym.

Rysunek 40 Dokładność drzewa klasyfikacyjnego w zależności od parametru złożoności



Z tablicy pomyłek dla drzewa klasyfikacyjnego (*Rysunek 41*) wynika, że model bardzo dobrze klasyfikuje gatunek operowy i klasyczny, jednak w odróżnieniu od modelu KNN wykazuje nieco większą tendencję do pomyłek się przy klasyfikacji muzyki operowej. Ponadto spośród wszystkich kategorii to właśnie opera jest najczęściej błędnie klasyfikowana jako muzyka klasyczna, jednak nigdy nie została pomylona z metalem.

Rysunek 41 Macierz pomyłek dla drzewa decyzyjnego

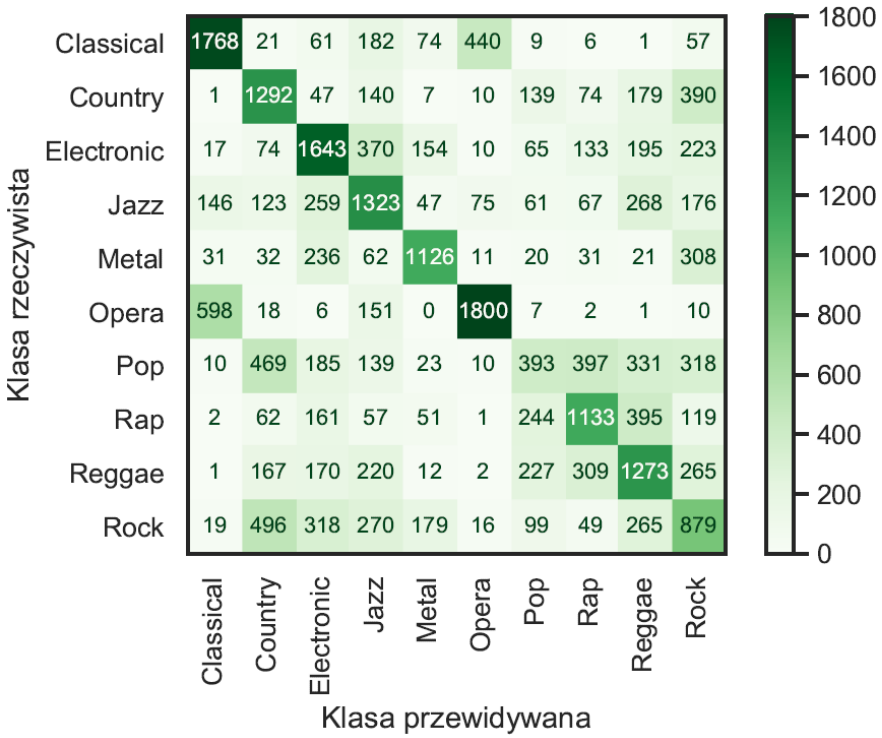


Tabela 11 Miary dokładności, precyzji oraz czułości drzewa decyzyjnego

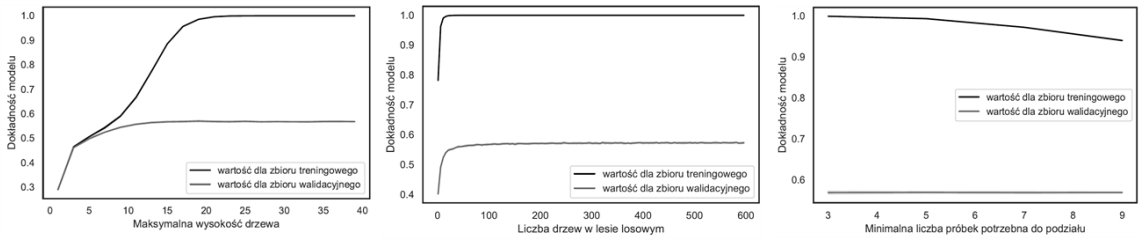
	Precyzja	**Czułość**	**Liczba utworów w danej klasie rzeczywistej**
Classical	0,68	0,68	2619
Country	0,47	0,57	2279
Electronic	0,53	0,57	2884
Jazz	0,45	0,52	2545
Metal	0,67	0,60	1878
Opera	0,76	0,69	2593
Pop	0,31	0,17	2275
Rap	0,51	0,51	2225
Reggae	0,43	0,48	2646
Rock	0,32	0,34	2590
Dokładność modelu	0,515		24534

Najsłabszą zdolność predykcyjną drzewo klasyfikacyjne posiada ponownie w przypadku gatunków rock oraz pop, charakteryzując utwory dla tych grup jako zupełnie inny gatunek. Oprócz tego model wykazuje ekstremalnie niską czułość dla popu, co wynika z trudności identyfikacji tego typu utworów. Klasyfikator ten w przypadku opery popełnia dużo mniej błędów podczas klasyfikacji niż KNN, na co wskazuje precyzja na poziomie 76%. Jednak procent dobrych decyzji, czyli poprawnych przypisań utworów do opery lub nie opery, jeśli był to inny gatunek, jest nieco niższy niż dla poprzedniego modelu. Miary dla pozostałych gatunków zawiera *Tabela 11*.

Las losowy

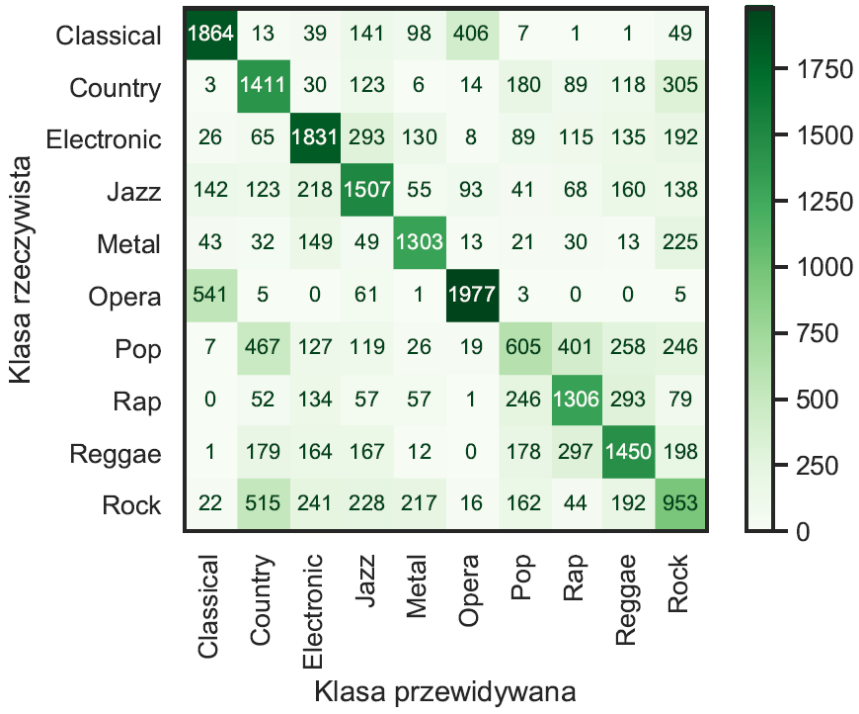
Podczas przygotowania modelu opartego na lesie losowym zbadany został wpływ wysokości pojedynczego drzewa, liczby estymatorów oraz minimalnej liczby obserwacji wymaganych przy podziale na dokładność modelu, co obrazuje kolejno *Rysunek 42*.

Rysunek 42 Krzywe walidacyjne dla lasu losowego



Już na tym etapie można stwierdzić, że model wygląda na nadmiernie dopasowany do danych uczących. Maksymalną wartością jaką można przyjąć jest 15. Niezależnie od tego jaka zostanie zastosowana liczba drzew wchodzących w skład lasu losowego oraz minimalna ilość obserwacji przy podziale drzewa, model ten osiąga taką samą dokładność. W związku z tym las losowy skonstruowany został dla 100 estymatorów z minimalną liczbą obserwacji równą 2.

Rysunek 43 Macierz pomyłek dla lasu losowego



Z macierzy pomyłek przedstawionej na *Rysunku 43* można odczytać, że las losowy najlepiej spośród dotychczasowych modeli radzi sobie z rozpoznawaniem gatunku operowego prezentując 1977 poprawnych przypisań. Las losowy również pozwolił na nieco mniej pomyłek w klasyfikacji, np. żaden utwór rapowy nie został przydzielony do muzyki klasycznej, czy reggae do operowej. W odniesieniu do poprzednich metod, piosenki pop uzyskały około 1,5 razy więcej trafionych predykcji.

Tabela 12 Miary dokładności, precyzji oraz czułości lasu losowego

	Precyzja	**Czułość**	**Liczba utworów w danej klasie rzeczywistej**
Classical	0,70	0,71	2619
Country	0,49	0,62	2279
Electronic	0,62	0,63	2884
Jazz	0,55	0,59	2545
Metal	0,68	0,69	1878
Opera	0,78	0,76	2593
Pop	0,39	0,27	2275
Rap	0,56	0,59	2225
Reggae	0,55	0,55	2646
Rock	0,40	0,37	2590
Dokładność modelu	0,58		24534

W *Tabeli 12* zebrano wartości precyzji oraz czułości predykcji poszczególnych gatunków. Utwory pop oraz rock są najczęściej klasyfikowane niepoprawnie jako inne gatunki, jednak dla tych kategorii model posiada minimalnie wyższy współczynnik czułości, zatem las losowy radzi sobie w tych przypadkach trochę lepiej. Dodatkowo można zauważyć, że przy klasyfikacji piosenek operowych las losowy w 76% poprawnie odróżnia utwory pochodzące z tego gatunku od innych oraz 78% przypisań do tego gatunku okazuje się być prawdziwych, co na tle pozostałych metod wypada najlepiej. Dość duża poprawa klasyfikacji występuje również w przypadku muzyki elektronicznej oraz jazzu. Ogólna dokładność modelu opartego na lesie losowym jest najwyższa spośród dotychczasowych modeli i wynosi 58%.

Sieci neuronowe

Dla sieci neuronowych zbudowane zostały 3 modele. Pierwszy z nich oparty na perceptronie wielowarstwowym cechuje się następującymi parametrami:

- liczba neuronów w warstwie ukrytej : 100,
- funkcja aktywacji : ReLU,
- algorytm optymalizacyjny : ADAM (ang. *Adaptive Moment Estimation*), oparty na stochastycznym gradiencie malejącym,
- liczba iteracji : 1000.

Drugi model skupiał się na splotowych sieciach neuronowych o strukturze funkcjonalnej, czyli łączonych warstwa po warstwie. Zawiera cztery warstwy ukryte z funkcją aktywacji ReLU:

- splotową (*Conv1D*) z 64 filtrami,
- łączącą (*MaxPooling1D*),
- spłaszczającą (*Flatten*),
- gęstą z 64 filtrami.

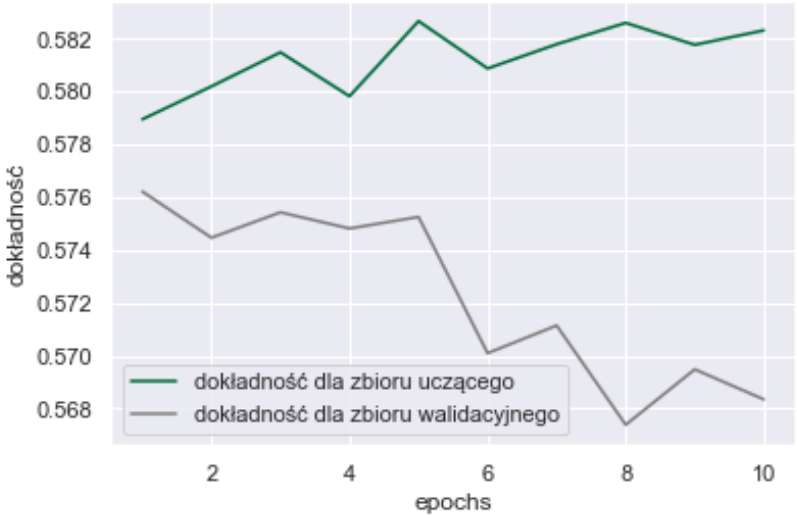
Na etapie kompilacji modelu dobrana została funkcja optymalizacji ADAM, dokładność jako miernik jakości oraz binarna entropia krzyżowa jako funkcja straty. Podczas procesu uczenia przyjęte zostały dwa podstawowe parametry:

- epochs=10, definiujący liczbę powtórzeń algorytmu na pełnym zestawie danych,
- batch_size=10, oznaczający liczbę obserwacji, przez które sieć musi przejść zanim zaktualizuje wewnętrzne parametry.

Innymi słowy, model przetworzy zbiór wejściowy w całości 10-krotnie, a co każde 10 obserwacji dokona aktualizacji parametrów modelu.

Przebieg działania algorytmu oraz dokładność modelu na zbiorze treningowym i walidacyjnym na poszczególnym etapie procesu uczenia został zwizualizowany na *Rysunku 44*. Skuteczność klasyfikacji na zbiorze walidacyjnym spadła w 5 powtórzeniu przy jednoczesnym minimalnym wzroście na danych uczących, a więc może to być oznaką przeuczenia.

Rysunek 44 Dokładność klasyfikacji w poszczególnych iteracjach dla sieci splotowych



Ostatni model został zbudowany w oparciu o głębokie sieci neuronowe, zawierające:

- 7 warstw gęstych o następujących liczbach filtrów: 64, 128, 256, 512, 256, 128, 64 wraz z normalizacją wag każdej z nich w celu przyspieszenia procesu uczenia
- 4 warstwy redukcji, w tym dwie z 25% oraz dwie z 35% prawdopodobieństwem usunięcia neuronu.

W tym przypadku model przetwarza zbiór 120 razy aktualizując parametry wewnętrzne co 200 próbek. Dokładność modelu wraz z kolejną iteracją wzrasta w przypadku zbioru uczącego, z kolei dla zbioru walidacyjnego skuteczność przestaje się znacząco zmieniać już przy 35 iteracji, a więc model ten ma potencjał do bycia przeuczonym, jednak następuje to dość powoli. Przebieg klasyfikacji za pomocą sieci głębokich przedstawia *Rysunek 45*.

Rysunek 45 Dokładność klasyfikacji w poszczególnych iteracjach dla sieci głębokich

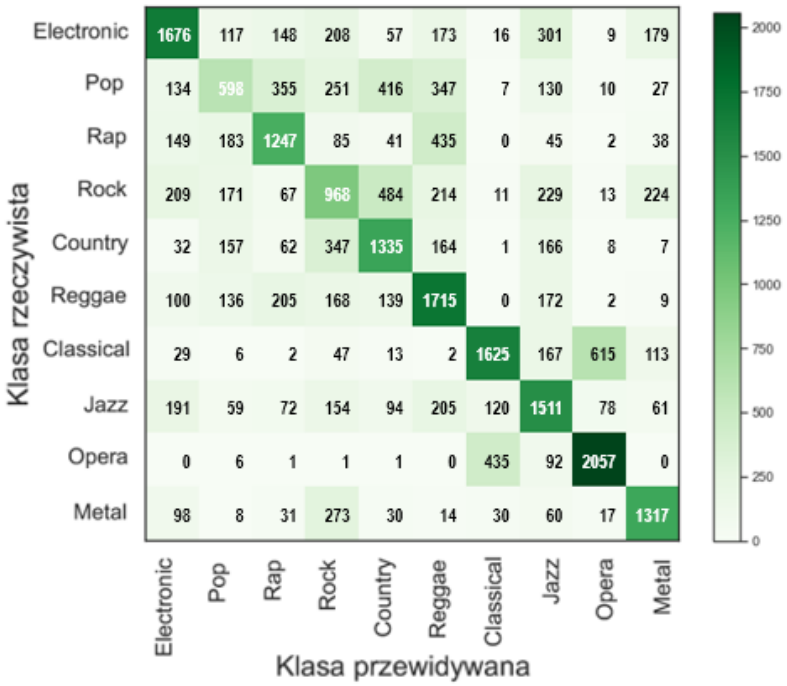
Warto zaznaczyć, że modele drugi i trzeci wytrenowane zostały przy użyciu zbioru danych zawierającego wszystkie cechy dźwiękowe, aby nie tracić informacji o danym utworze. Wystarczająco głębokie sieci powinny radzić sobie z wysoko skorelowanymi zmiennymi, ponieważ są zdolne do wykrywania dodatkowych zależności.

Wstępne porównanie dokładności trzech wersji modeli opartych na sieciach neuronowych zawarte zostało w *Tabeli 13* i na tej podstawie zdecydowano się przeanalizować jedynie ostatni model ze względu na najlepszą trafność klasyfikacji.

Tabela 13 Porównanie dokładności sieci neuronowych

	dokładność modelu
sieć MLP	55,90%
sieć splotowa	56,78%
sieć gęsta	57,26%

Rysunek 46 Macierz pomyłek dla sieci neuronowych



Jak pokazano na *Rysunku 46*, po raz kolejny zmiana modelu spowodowała wzrost liczby trafnych predykcji dla gatunku opera. Nie znaleziono przypadku, w którym sieci neuronowe pomyliłyby muzykę reggae oraz rap z klasyczną czy operę z metalem, jednak zdarzały się przypadki odwrotne- utwór metalowy kilka razy trafił do kategorii

operowej, a muzyka klasyczna do rapu oraz reggae. Model ten również nie radzi sobie dobrze z klasyfikacją popu oraz rocka.

Tabela 14 Miary dokładności, precyzji oraz czułości sieci neuronowej

	Precyzja	**Czułość**	**Liczba utworów w danej klasie rzeczywistej**
Electronic	0,64	0,58	2884
Pop	0,41	0,26	2275
Rap	0,57	0,56	2225
Rock	0,39	0,37	2590
Country	0,51	0,59	2279
Reggae	0,52	0,65	2646
Classical	0,72	0,62	2619
Jazz	0,53	0,59	2545
Opera	0,73	0,79	2593
Metal	0,67	0,70	1878
Dokładność modelu	0,57		24534

Spośród wszystkich modeli sieci neuronowe w przypadku muzyki pop cechują się najlepszą precyzją 41%, czyli model wskazał, że 1441 piosenek pochodzi z gatunku pop, jednak poprawnie zaklasyfikował jedynie 598 utworów. Na tym przykładzie czułość klasyfikatora dla popu wynosi zaledwie 26% przy 2275 przykładach w zbiorze testowym. Z kolei do gatunku opera przypisanych zostało 2811 piosenek, a więc więcej niż występuje rzeczywiście w zbiorze, co w efekcie daje precyzję równą około 73%. To właśnie dla tego gatunku sieć neuronowa osiąga maksymalną zdolność predykcji. Model ten poprawnie klasyfikuje dużą ilość utworów operowych w odniesieniu do ilości przykładów w danych pochodzących rzeczywiście z tego gatunku. Miary te dla pozostałych kategorii zebrane zostały w *Tabeli 14*.

Stosunek poprawnie przydzielonych gatunków do liczby wszystkich utworów w zbiorze w tym przypadku wynosi 57%.

4.4 Porównanie jakości modeli klasyfikacyjnych

Powstałe modele klasyfikacyjne nie wyróżniają się bardzo dużą dokładnością. Klasyfikator K-najbliższych sąsiadów oraz pojedyncze drzewo klasyfikacyjne osiągnęły maksymalną trafność oscylującą w okolicy 51%, natomiast las losowy oraz sieć neuronowa 57%. W przypadku gatunku operowego, każdy z modeli cechuje się najwyższym odsetkiem prawidłowych klasyfikacji w odniesieniu do wszystkich przypisań utworów do tej klasy. Należy tutaj wyróżnić model oparty na lesie losowym, ponieważ osiągnął precyzję 78%. Z kolei najmniejsza zdolność do wychwycenia rzeczywistego gatunku występowała w większości przypadków dla muzyki pop, co może oznaczać, że modele klasyfikowały piosenki pop częściej do innych kategorii. Jedynym wyjątkiem jest model sieci neuronowych, ponieważ najczęściej myli utwory rockowe z innymi gatunkami. Klasyfikator KNN w precyzji predykcji wypada najsłabiej. Najwięcej trafnych przypisań do danego gatunku w stosunku do wszystkich utworów w zbiorze testowym modele wykazywały również w przypadku opery oraz najmniej w przypadku popu. Sieci neuronowe okazały się mieć najwyższą czułość, osiągając wartość 79%, natomiast drzewo klasyfikacyjne najniższą równą około 17%.
Dodatkowo przy zestawieniu maksymalnych precyzji, model *K*-najbliższych sąsiadów oraz sieci neuronowych ma tendencję do klasyfikowania większej liczby utworów operowych niż występuje w rzeczywistym zbiorze jako gatunek operowy, o czym świadczy współczynnik precyzji niższy niż miara czułości. Opisane zależności zawarte zostały w *Tabeli 15*.

Tabela 15 Porównanie modeli klasyfikacyjnych

Model	Dokładność	Największa precyzja	Największa czułość	Najmniejsza precyzja	Najmniejsza czułość
KNN	0,5108	0,68 (Opera)	0,75 (Opera)	0,29 (Pop)	0,2 (Pop)
Drzewo klasyfikacyjne	0,5148	0,76 (Opera)	0,69 (Opera)	0,31 (Pop)	0,17 (Pop)
Las losowy	0,5791	0,78 (Opera)	0,76 (Opera)	0,39 (Pop)	0,27 (Pop)
Głębokie sieci neuronowe	0,5726	0,73 (Opera)	0,79 (Opera)	0,39 (Rock)	0,26 (Pop)

Skonstruowane modele słabo sprawdzają się przy klasyfikacji większej liczby gatunków, co jest prawdopodobnie spotęgowane dość zbieżnymi wartościami parametrów utworów. Z uwagi na to, że przykładowo muzyka pop czy rock cieszy się

bardzo dużą popularnością, bardzo często utwory są określane tymi gatunkami, mimo tego, że posiadają cechy charakterystyczne dla zupełnie innego typu muzyki.

Jako, że najlepiej w zestawieniu modeli wypadł las losowy, przeprowadzono klasyfikację binarną dla gatunku pop, aby sprawdzić w jakim stopniu model będzie w stanie wychwycić utwory z tej kategorii na tle innych gatunków. Przyjęte zostały dwie klasy: *Pop* oraz *Others*. Ilość danych została wyrównana dla obydwóch klas, aby model miał równe szanse.

Model klasyfikacyjny rozróżniający pop od innych gatunków okazuje się być bardziej skuteczny niż wieloklasowy. Zgodnie z miarami z *Tabeli 16* oraz macierzą pomyłek przedstawioną na *Rysunku 47*, dokładność tego klasyfikatora to 75%. Posiada on dużą zdolność do prawidłowego rozróżniania utworów pop spośród innych (84%), jednak na takim zbiorze przykładów w wielu przypadkach fałszywie przypisuje odmienny gatunek do popu, stąd niższa wartość precyzji wynosząca 71%.

Rysunek 47 Macierz pomyłek dla lasu losowego w problemie klasyfikacji binarnej

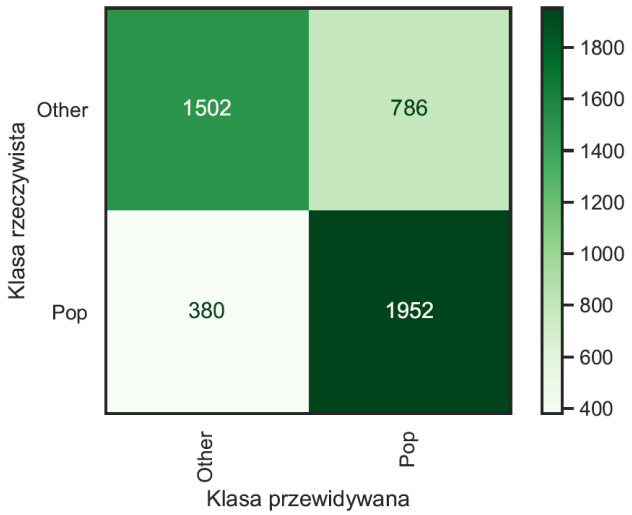
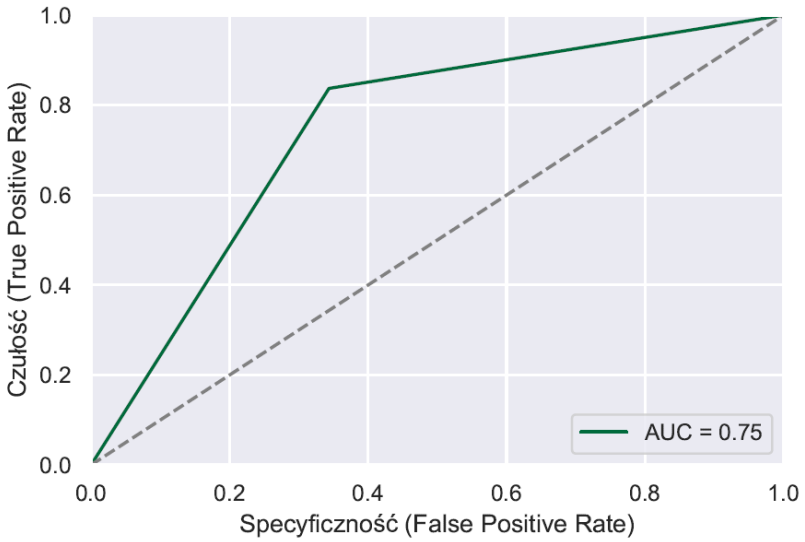


Tabela 16 Miary dokładności, precyzji i czułości dla modelu opartego na lesie losowym w klasyfikacji binarnej

	Precyzja	**Czułość**	**Liczba utworów w danej klasie rzeczywistej**
Other	0,80	0,66	2288
Pop	0,71	0,84	2332
Dokładność modelu	0,75		4620

Model ten można ocenić dodatkowo za pomocą krzywej ROC oraz pola pod wykresem, nazywanego miarą AUC, co zawarte zostało na *Rysunku 48*. Wykres ten przedstawia zależność pomiędzy odsetkiem prawidłowych klasyfikacji utworów do gatunku pop oraz odsetkiem błędnych przypisań do tego gatunku. Współczynnik AUC wynosi w tym przypadku 75%, co jest dość dobrym wynikiem i świadczy o dużej mocy diagnostycznej modelu przy rozróżnianiu gatunków.

Rysunek 48 Krzywa ROC dla lasu losowego w klasyfikacji binarnej



W związku z tym, można powiedzieć, że tego typu klasyfikacja będzie bardziej użyteczna w celu odfiltrowania na przykład niechcianych gatunków z list odtwarzania niż klasyfikacja wielu gatunków.

Podsumowanie

Praca ta miała na celu zestawienie algorytmów wykorzystywanych w klasyfikacji, która jest nieodłączną częścią systemów rekomendacyjnych znanych serwisów udostępniających muzykę. Część badawcza podzielona została na część wstępnej analizy skupień opartej na grupowaniu podobnych utworów muzycznych oraz właściwą klasyfikację gatunków. W analizie wykorzystane zostały dane dotyczące cech dźwiękowych utworów udostępniane przez serwis Spotify.

Dwie metody, jakie zostały zastosowane do grupowania dostępnego zbioru danych, czyli algorytm K-średnich i hierarchiczny Warda pozwoliły na otrzymanie nieco odmiennych wyników w postaci skupień piosenek o podobnym charakterze. Pomimo występowania aż 10 różnych gatunków w danych, możliwe było wyodrębnienie nie więcej niż trzech list utworów. Metoda K-średnich wyznaczyła 3 playlisty: neutralną, rytmiczną oraz poważną, jednak pierwsza grupa okazała się być bardzo dużym skupiskiem piosenek z niemal wszystkich analizowanych gatunków. Przy grupowaniu hierarchicznym można powiedzieć, że grupa neutralna i rytmiczna otrzymane przy poprzednim sposobie klastrowania została połączona w jedną. W związku z tym, ostatnia lista piosenek okazała się być najbardziej wyraźna i na tej podstawie można się spodziewać, że utwory należące do muzyki klasycznej i operowej będą najlepiej rozróżniane w dokładnej klasyfikacji. W grupowaniu wyboru cech dokonano jedynie na podstawie korelacji, jednak warto byłoby pomyśleć o przeprowadzeniu analizy głównych składowych (PCA) w celu wyeliminowania niektórych zmiennych. Wtedy wyniki mogłyby być nieco inne.

Cechy audio, które głównie wpływały na charakterystykę poszczególnej listy odtwarzania to głównie poziom występowania wokalu w utworze wraz z liczbą wykrytych słów, akustyczność i instrumentalność, a także nastrój danej ścieżki.

Kolejna część pracy stanowiła budowę i porównanie modeli klasyfikacyjnych opartych na czterech algorytmach: K-najbliższych sąsiadów, drzewach klasyfikacyjnych, lesie losowym oraz sieciach neuronowych. Otrzymane klasyfikatory nie wykazały satysfakcjonujących rezultatów, pomimo prób dostosowania parametrów poprzez poszukiwanie najbardziej optymalnych, osiągając dokładność nieco ponad 50%. Zdecydowanie jest to problem wielu klas występujących w danych, skutkując częstym popełnianiem błędów. Dodatkowo wybrane gatunki mogły być za mało zróżnicowane, aby wyraźnie charakteryzować kategorię muzyczną. Ze słabą zdolnością predykcyjną

modeli wiąże się również dobór cech charakteryzujących utwory. Być może nie są one wystarczające do jednoznacznego określenia gatunków. Spotify oferuje również dane dotyczące przykładowo bitów, przesunięcia taktu, podziału utworu na sekcje, czy segmenty, które najczęściej posiadają spójne cechy dźwiękowe (np. refren). Prawdopodobnie dodanie takich informacji poprawiłoby trafność modeli, ponieważ uwzględniałyby one dodatkowe czynniki.

Kolejną kwestią, która mogła mieć wpływ na jakość klasyfikacji jest liczba danych, która szczególnie w przypadku sieci neuronowych ma bardzo duże znaczenie. Pomimo głównej zalety głębokich sieci neuronowych, jaką jest wykrywanie nawet najmniejszych zależności w zbiorze danych, model nie był zadowalający. Ponadto zastosowanie splotowych sieci, które sprawdzają się bardzo dobrze przy wykrywaniu zależności między blisko położonymi obiektami głównie na obrazach, prawdopodobnie nie jest zbyt zasadne w przypadku danych tabelarycznych, w których ustawienie kolejności cech nie ma znaczenia.

Przy metodzie KNN nie została rozróżniona istotność poszczególnych zmiennych, a więc jest tu miejsce na wprowadzenie dodatkowej analizy tych czynników i dokonanie predykcji wykorzystując jedynie najbardziej istotne z nich. W odróżnieniu od tej techniki klasyfikacji, lasy losowe radzą sobie z tego typu problemem decydując o tym, które cechy uwzględnić przy predykcji. Lasy losowe w niniejszej analizie okazały się być najskuteczniejszą z metod, ze względu na dodatkowy element losowości, który pozwala na osiągnięcie niezależności zmiennych i tym samym zmniejszenie ryzyka przeuczenia modelu. Pojedyncze drzewo klasyfikacyjne nie było w tym przypadku wystarczające ze względu na niską stabilność.

Mając do dyspozycji jedynie cechy audio dostępne na Spotify, bardziej sensownym modelem byłby taki, który rozróżnia jeden gatunek na tle innych, udostępniając bardziej klarowną rekomendację z uwagi na wyraźniejsze wydobycie charakteru poszczególnych zmiennych objaśniających.

Podsumowując, modele skonstruowane na potrzeby tej analizy nie pozwoliłyby na jednoznaczną klasyfikację gatunków, ponieważ trafnością na poziomie 50% nie wnosiłyby niczego istotnego do systemu rekomendacji. Z tego powodu serwis Spotify w swoich algorytmach uwzględnia z pewnością dużo więcej dodatkowych czynników poprawiających filtrowanie kategorii utworów, a co za tym idzie, trafniejsze polecenia nowych utworów na podstawie upodobań użytkowników.

Literatura

[1] Balicki A., *Statystyczna analiza wielowymiarowa i jej zastosowania społeczno-ekonomiczne*, Wydawnictwo Uniwersytetu Gdańskiego, 2013

[2] Gatnar E., Walesiak M., *Statystyczna analiza danych z wykorzystaniem program R*, Wydawnictwo naukowe PWN, Warszawa, 2012

[3] Górecki T., Krzyśko M., Skorzybut M., Wołyński W., *Systemy uczące się*, Wydawnictwo Naukowo-Techniczne Warszawa, 2008

[4] Hastie T., James G., Witten D., Tibshirani R., *An Introduction to Statistical Learning*, Springer, New York, 2013

[5] Hastie T., Tibshirani R., Friedman J., *The Elements of Statistical Learning. Data Mining, Inference, and Prediction. Second Edition*, Springer, 2009

[6] Koronacki J., Ćwik J., *Statystyczne systemy uczące się. Wydanie drugie*, Akademicka Oficyna Wydawnicza EXIT, 2015

[7] Mazur D., *Metody grupowania i ich implementacja do eksploracji danych postaci symbolicznej*, Praca doktorska, 2005

[8] Schafer J.B. , Frankowski D., Herlocker J , Sen S., *Collaborative Filtering Recommender Systems*, Conference Paper on ResearchGate, 2007

[9] Tadeusiewicz R., Gąciarz T., B.Borowik, B.Leper, *Odkrywanie właściwości sieci neuronowych*, Polska Akademia Umiejętności, 2007

[10] Walesiak M., *Przegląd formuł normalizacji wartości zmiennych oraz ich własności w statystycznej analizie wielowymiarowej*, Przegląd statystyczny R. LXI - Zeszyt 4, 2014

[11] Zocca V., Spacagna G., Slater D., Roelants P., *Deep Learning. Uczenie głębokie z językiem Python. Sztuczna inteligencja i sieci neuronowe*, Helion, 2018

Źródła internetowe

[1] Giacaglia G., *Spotify's Recommendation Engine*, Medium, 2019, https://medium.com/datadriveninvestor/behind-spotify-recommendation-engine-a9b5a27a935

[2] Gibiansky A., *Convolutional Neural Networks*, 2014, https://andrew.gibiansky.com/blog/machine-learning/convolutional-neural-networks/

[3] Guneysu U., *How Is Spotify's Thriving Recommendation System Becoming A New Advertising Platform*, Medium, 2019, https://medium.com/swlh/how-is-spotifys-thriving-recommendation-system-becoming-a-new-advertising-platform-a2b97ffe2012

[4] Harańczyk G., *Krzywe ROC, czyli ocena jakości klasyfikatora i poszukiwanie optymalnego punktu odcięcia*, StatSoft, 2010, https://media.statsoft.pl/_old_dnn/downloads/krzywe_roc_czyli_ocena_jakosci.pdf

[5] Horzyk A., *Uczenie głębokie i głębokie sieci neuronowe*, Wydział EAIIB AGH, 2018, http://home.agh.edu.pl/~horzyk/lectures/miw/MIW-DL.pdf

[6] Mwiti D., *Convolutional Neural Networks: An Intro Tutorial*, Medium, 2018, https://heartbeat.fritz.ai/a-beginners-guide-to-convolutional-neural-networks-cnn-cf26c5ee17ed

[7] Rogalewski P., *AI dla każdego. Część 3*, Czasopismo Zabezpieczenia, 2019, https://www.zabezpieczenia.com.pl/nowe-technologie/ai-dla-kazdego-czesc-3

[8] Saha S., *A comprehensive guide to convolutional neural networks*, Medium, 2018 https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[9] *Analiza danych pomiarowych: Analiza skupień,* AGH Inżynieria biomedyczna, 2014/2015, http://home.agh.edu.pl/~mmd/_media/dydaktyka/adp/analiza_skupien.pdf

[10] *Jak ocenić jakość i poprawność modeli klasyfikacyjnych? Część 1 – Wprowadzenie*, Algolytics, https://algolytics.pl/tutorial-jak-ocenic-jakosc-i-poprawnosc-modeli-klasyfikacyjnych-czesc-1-wprowadzenie/

[11] *Jak ocenić jakość i poprawność modeli klasyfikacyjnych? Część 3 – Confusion Matrix*, Algolytics, https://algolytics.pl/jak-ocenic-jakosc-i-poprawnosc-modeli-klasyfikacyjnych-czesc-3-confusion-matrix/

[12] *K-means Cluster Analysis*, UC Business Analytics R Programming Guide, 2017, https://uc-r.github.io/kmeans_clustering

[13] *K-najbliższych sąsiadów*, StatSoft, https://www.statsoft.pl/textbook/stathome_stat.html?https%3A%2F%2Fwww.statsoft.pl%2Ftextbook%2Fstknn.html

[14] *Konwolucyjne sieci neuronowe 3: overfitting*, AI Geek Programmer, 2020, https://aigeekprogrammer.com/pl/konwolucyjne-sieci-neuronowe-tutorial-czesc-3/

[15] *Sieci neuronowe*, StatSoft, https://www.statsoft.pl/textbook/stathome_stat.html?https%3A%2F%2Fwww.statsoft.pl%2Ftextbook%2Fstneunet.html

[16] *Spotify for developers: Get Audio Features for a Track*, Spotify AB, 2020, https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/

[17] *Spotify for developers: Web API*, Spotify AB, 2020, https://developer.spotify.com/documentation/web-api/

[18] *The history of music distribution*, MN2S, 2020, https://mn2s.com/news/label-services/the-history-of-music-distribution/

[19] *Understanding Spotify: Making Music Through Innovation*, Goodwater, 2018, https://www.goodwatercap.com/thesis/understanding-spotify

[20] *Zasięg (TPR – czułość / TNR – specyficzność) i precyzja (PPV / NPV) – czyli ocena jakości klasyfikacji (część 2)*, Mathspace, http://mathspace.pl/matematyka/ocena-jakosci-klasyfikacji-czesc-2/

Spis tabel

Spis ilustracji