

Studium Magisterskie

Kierunek ANALIZA DANYCH – BIG DATA

Specjalność -

Imię i nazwisko autora

Aleksandra Gomółka

Nr albumu 123034

Analiza sentymentu w mediach społecznościowych z wykorzystaniem technik uczenia maszynowego

Praca magisterska

pod kierunkiem naukowym

dr Mariusz Rafało

Instytut

Instytut Informatyki i Gospodarki Cyfrowej

Warszawa 2024

Spis treści

[Wstęp 4](#_Toc170308149)

[1 Analiza Sentymentu 6](#_Toc170308150)

[1.1 Wprowadzenie do Analizy Sentymentu 6](#_Toc170308151)

[1.2 Przegląd narzędzi i technologii w analizie Sentymentu 15](#_Toc170308152)

[1.3 Podstawy Przetwarzania Języka Naturalnego 20](#_Toc170308153)

[1.4 Potencjalne wyzwania i ograniczenia 25](#_Toc170308154)

[2 Fundamenty teoretyczne metod Uczenia Maszynowego w Analizie Sentymentu 27](#_Toc170308155)

[2.1 Uczenie Nadzorowane 29](#_Toc170308156)

[2.1.1 Regresja Logistyczna 30](#_Toc170308157)

[2.1.2 Metoda K - Najbliższych Sąsiadów 33](#_Toc170308158)

[2.1.3 Drzewo Decyzyjne 35](#_Toc170308159)

[2.2 Uczenie Głębokie 38](#_Toc170308160)

[2.2.1 Sztuczne Sieci Neuronowe 42](#_Toc170308161)

[2.2.2 Sztuczne Sieci Neuronowe Konwolucyjne 43](#_Toc170308162)

[2.2.3 Rekurencyjne Sieci Neuronowe 43](#_Toc170308163)

[2.3 Algorytmy Przetwarzania Języka Naturalnego 52](#_Toc170308164)

[2.3.1 Przetwarzanie wstępne i reprezentacja tekstu 52](#_Toc170308165)

[2.3.2 Redukcja wymiarowości i jej wpływ na analizę tekstu 56](#_Toc170308166)

[2.3.3 Metody Analizy Sentymentu 57](#_Toc170308167)

[2.3.4 Graficzne przedstawienie danych w Przetwarzaniu Języka Naturalnego 59](#_Toc170308168)

[2.4 Regularyzacja 60](#_Toc170308169)

[2.5 Optymalizacja 65](#_Toc170308170)

[3 Analiza Sentymentu w mediach społecznościowych 69](#_Toc170308171)

[3.1 Charakterystyka i przygotowanie danych 69](#_Toc170308172)

[3.2 Metody redukcji wymiarów w NLP 78](#_Toc170308173)

[3.3 Analiza sentymentu z użyciem narzędzi VADER i NLTK 80](#_Toc170308174)

[3.4 Budowa i optymalizacja modeli klasyfikacyjnych 81](#_Toc170308175)

[3.4 Interpretacja i dyskusja nad wynikami Analizy Sentymentu 87](#_Toc170308176)

[3.5 Wymagania techniczne i sprzętowe 94](#_Toc170308177)

[Zakończenie 96](#_Toc170308178)

[Bibliografia 98](#_Toc170308179)

[Spis rysunków 105](#_Toc170308180)

[Spis tabel 108](#_Toc170308181)

[Załączniki 108](#_Toc170308182)

[Streszczenie 109](#_Toc170308183)

# Wstęp

W dynamicznym krajobrazie ery cyfrowej media społecznościowe stały się kluczową platformą globalnej wymiany myśli, opinii i emocji, rewolucjonizując komunikację na niespotykaną dotąd skalę. Powszechna dostępność urządzeń elektronicznych umożliwiła codzienne interakcje na całym świecie, skutecznie przekraczając granice geograficzne i bariery językowe. Ogromna ilość informacji generowanych codziennie przez użytkowników, w połączeniu z ich powszechną dostępnością, aktualnością dyskusji i różnorodnością platform do dzielenia się doświadczeniami, przekształciła media społecznościowe w bogate źródło danych tekstowych. Próba zrozumienia, w jaki sposób ludzie wyrażają swoje opinie w formie pisemnej, nie została hermetycznym wyzwaniem akademickim, ale ma również swój ogromny udział w biznesie, polityce, rynkach finansowych, marketingu i wielu innych. Rozważanie nad tym tematem stanowi główny cel analizy sentymentu – metody, która zajmuje się przetwarzaniem języka naturalnego. W następstwie tego podejścia, powstaje proces, który pozwala na klasyfikację nastroju, który dominuje w tekstach.

Niniejsza praca koncentruje się na wykorzystaniu zaawansowanych technik uczenia maszynowego do przetwarzania języka naturalnego w szczególności koncentrując się na badaniu sentymentu postów w mediach społecznościowych, aby zrozumieć opinię publiczną i reakcje emocjonalne wobec różnych marek, celebrytów, miast, instytucji. Dane zostały pierwotnie zebrane przez opinmind.com, która była komercyjną wyszukiwarką blogów internetowych. Źródło to zapewnia, że dane pochodzą z naturalnych wypowiedzi użytkowników. Zestaw danych UMICH S1650 został uprzednio wykorzystany w pracach naukowych, co pozwala na porównanie otrzymanych wyników oraz ocenę jakości, innowacyjności proponowanych rozwiązań. Różnorodność tematów w danych umożliwiła trenowanie modeli zdolnych do rozpoznawania sentymentu w różnych kontekstach, niezależnie od specyfiki tematycznej. Trenowanie modeli na informacjach ograniczonych do jednej tematyki może prowadzić do nadmiernego dopasowania, a szerokie spektrum daje możliwość zastosowania w narzędziach monitoringu czy rekomendacji w szeroko zakrojonych przeznaczeniach. Celem badania jest skategoryzowanie sentymentu jako pozytywnego lub negatywnego, aby uzyskać wgląd w ogólne postrzeganie i preferencje opinii publicznej. Zadanie to jest problemem klasyfikacji tekstu, w którym celem jest zbudowanie modelu, który może dokładnie przewidzieć sentyment komentarzy na podstawie danych treningowych. Spostrzeżenia mogą być wykorzystane do głębszego zrozumienie poglądów oraz zarządzania reputacją celebrytów lub dostosowaniu ich publicznych wystąpień do widowni. Drużyny sportowe mogą wykorzystywać analizę sentymentu do poprawy doświadczeń fanów, czy tego w jak sposób postrzegane są wyniki zespołu, a przez to zwiększenia ich wsparcia. Wgląd w nastroje mieszkańców miast może pomóc w ulepszaniu usług publicznych, infrastruktury, a także obraniu nowych strategii turystycznych. Dążeniem analizowania posiadanych danych dotyczących ludzkich opinii jest dostarczenie przydatnego wglądu dla interesariuszy, pomagając im w podejmowaniu świadomych decyzji, aby ulepszyć produkty, usługi, co ma w dłuższym okresie przynieść pochlebne postrzeganie przez odbiorców.

Działania podjęte w tej pracy wpisują się w szerokie ramy współczesnych badań interdyscyplinarnych, łączących w sobie elementy informatyki, lingwistyki oraz socjologii. W ten sposób zaoferowano praktyczne podejście do analizy, przetwarzania i klasyfikacji zbiorów danych tekstowych. W ramach obranego podejścia, szczególną uwagę skupiono na skomponowaniu, a następnie ocenie modeli uczenia maszynowego, które mają zdolność do rozróżniania przekazanych emocji występujących we wpisach. Ponadto, praca poszerzona jest o eksplorację wielorakich technik przetwarzania, w tym na wektoryzację słów, metody redukcji wymiarów oraz wybór optymalnych parametrów.

Podsumowując, niniejsza praca nie tylko podkreśla znaczącą rolę analizy sentymentu w zrozumieniu publicznego dyskursu w mediach społecznościowych, ale także pokazuje, w jaki sposób kształtuje się opinia publiczna i reakcje emocjonalne wobec różnych marek, celebrytów, miast oraz instytucji. Analiza nastrojów jest już szeroko stosowana przez platformę Rotten Tomatoes, gdzie agreguje się recenzje filmów od krytyków i widzów, aby określić ogólny sentyment każdej uwagi. Pozwala im to zapewnić wynik "Tomatometru", który reprezentuje procent pozytywnych recenzji. Na przykład, film z wysokim wynikiem Tomatometru jest prawdopodobnie pozytywnie postrzegany zarówno przez krytyków, jak i widzów. Pomaga to użytkownikom zdecydować, które filmy obejrzeć w oparciu o zagregowane dane dotyczące nastrojów. Firma Disney wykorzystuje analizę nastrojów do pomiaru zaangażowania odbiorców na różnych platformach, w tym w mediach społecznościowych, blogach, forach. Pomaga to firmie zrozumieć, jak ich produkcje są odbierane na całym świecie. Na przykład, jeśli nowy serial animowany generuje pozytywne opinie na temat jego fabuły, czy postaci, Disney może wykorzystać tę wiedzę do planowania kontynuacji lub spin-offów. Miasto Los Angeles wykorzystuje analizę nastrojów do zrozumienia opinii publicznej na temat bezdomności i skuteczności powiązanych polityk miejskich. Analiza ta ujawniła silne poparcie społeczne dla bardziej humanitarnych oraz skutecznych rozwiązań w zakresie bezdomności. Wyrażone komentarze doprowadziły do wdrożenia nowych programów koncentrujących się na zapewnieniu stałych mieszkań socjalnych, a także usług w zakresie zdrowia psychicznego. Zmiany te przyczyniły się do poprawy warunków życia osób w kryzysie bezdomności, zwiększając społeczne poparcie dla działań podejmowanych przez miasto. Tego typu badania prowadzone przez różne instytucje zapowiadają powstanie coraz bardziej złożonych i wyrafinowanych systemów. Postępy w tej dziedzinie zwiększą naszą zdolność do uzyskiwania zniuansowanego wglądu w emocje osadzone w piśmie, a także do generowania naturalnych, podobnych do ludzkich reakcji.

# 1 Analiza Sentymentu

## 1.1 Wprowadzenie do Analizy Sentymentu

Analiza sentymentu (ang. Sentiment Analysis), znana również jako przetwarzanie opinii, jest procesem automatycznego rozpoznawania i klasyfikacji emocji wyrażonych w tekście, stanowiącym kluczowy element w dynamicznie rozwijającym się świecie cyfrowym, gdzie dane tekstowe są nieustannie generowane (Liu, 2012). Głównym celem tego badania jest zrozumienie postawy autora wypowiedzi, co jest niezmiernie istotne w kontekście szybko zmieniających się trendów i dynamiki społecznej w portalach internetowych. Technika ta pozwala nie tylko na identyfikowanie ogólnego tonu komunikatu - pozytywnego, negatywnego czy neutralnego (rysunek 1) - ale również na odkrywanie subtelniejszych niuansów emocjonalnych, takich jak złość, radość czy smutek (Pang & Lee, 2008). To umożliwia głębsze zrozumienie ludzkich reakcji i zachowań, co ma istotne znaczenie zarówno w sferze społecznej, jak i biznesowej.



Rysunek 1. Przykłady klasyfikacji sentymentu w opinii o filmach. Źródło: opracowanie własne

Zastosowanie analizy sentymentu może być doskonale zilustrowane na przykładzie prostego, indywidualnego stwierdzenia, które można napotkać w mediach społecznościowych lub recenzjach internetowych. Załóżmy, że użytkownik aplikacji publikuje następujący post: "*Jestem rozczarowany nową aktualizacją aplikacji. Spowodowała wiele błędów i spowolniła działanie mojego telefonu.*". W tym prostym stwierdzeniu, analiza sentymentu pozwala na jednoznaczne sklasyfikowanie ogólnego tonu wypowiedzi jako negatywnego. Słowa takie jak "*rozczarowany*", "*błędy*" i "*spowolniła*" bezpośrednio wskazują na negatywne emocje użytkownika względem ostatniej aktualizacji aplikacji. Tego typu klarowne identyfikowanie znaczenia tekstu jest szczególnie przydatne dla firm technologicznych, które monitorują opinie użytkowników o ich produktach lub usługach. Umożliwia to szybkie reagowanie na problemy i pracę nad ich rozwiązaniem, co jest kluczowe dla utrzymania wysokiej jakości doświadczenia klienta, co ukazano na rysunku 2.



Rysunek 2. Schemat przepływu procesu obsługi klienta w zakresie zgłaszania i rozwiązywania usterek w oprogramowaniu. Źródło: opracowanie własne

Rozwój technologii informatycznych i cyfryzacja społeczeństwa spowodowały, że analiza sentymentu zyskała na znaczeniu, stając się nieodzownym narzędziem w wielu dziedzinach, od marketingu po monitorowanie opinii publicznej (Bollen, Mao, & Zeng, 2011). W dzisiejszym świecie, gdzie opinie są swobodnie wyrażane na wielu platformach społecznościowych, blogach, czy w recenzjach produktów, analiza sentymentu staje się nie tylko narzędziem badawczym, ale także strategicznym zasobem dla spółek i organizacji. Pozwala ona na wychwycenie subtelnych zmian w percepcji marki, produkcie czy usłudze, co jest nieocenione dla decydentów biznesowych. Dobrym przykładem jest analiza opinii wykonywana przez serwis Netflix. Spośród wielu pojawiających się platform do przesyłania filmów, Netflix jest tą, która ma najwięcej subskrybentów w porównaniu do innych (Abelard & Sibaroni, 2021). Platforma ta porównuje nastroje odbiorców dotyczące różnych filmów i na tej podstawie rekomenduje treści o podobnej tematyce lub kategorii. Grupując treści na podstawie analizy sentymentu, Netflix może oferować konkretne rekomendacje atrakcyjne dla widzów, którzy lubią określone nastroje lub style opowiadania historii (wonderflow.ai, 2024). Aby to osiągnąć i zyskać na konkurencyjności używają oni m. in. rozproszonego przetwarzania danych przez Apache Spark czy Flink, które jest kluczowe dla efektywnego przetwarzania ogromnych ilości danych. Natomiast w celu zbudowania i trenowania zaawansowanych modeli analizy sentymentu, które potrafią zrozumieć oraz kategoryzować emocje klientów na podstawie danych tekstowych stosują technologie takie jak: TensorFlow, Keras czy PyTorch (jobs.netflix.com, 2024). Zastosowanie analizy sentymentu w biznesie otwiera nowe możliwości w obrębie personalizacji oferty, zwiększenia satysfakcji klientów i optymalizacji strategii marketingowych. Firmy, które potrafią efektywnie interpretować emocje zyskują cenne wskazówki, które mogą przekuć w działania prowadzące do wzrostu lojalności kupujących, poprawy reputacji marki, jak również w konsekwencji zwiększenia przychodów.

Przedsiębiorstwo Amazon, na przykład, wykorzystuje analizę nastrojów do monitorowania oraz zarządzania recenzjami w swoim sklepie internetowym, co pozwala firmie szybko reagować na wszelkie negatywne opinie, a przez to zwiększać zadowolenie klientów. Przewodniki techniczne AWS opisują krok po kroku metody konfigurowania potoków analizy nastrojów za pomocą Amazon Comprehend, prezentując jego zastosowanie w czasie rzeczywistym do zarządzania opiniami klientów (aws.amazon.com, 2024). Firma założona przez Jeffa Bezosa zbiera recenzje z wysokim wynikiem negatywnych opinii i automatycznie oznacza je, jako wymagające dalszej uwagi. Dalej są one wysyłane do zespołu obsługi klienta, który priorytetowo traktuje odpowiadanie na te recenzje, aby szybko rozwiązać problemy klientów (dzone.com, 2024).

Forma publikowania krótkich, zwięzłych wpisów w Internecie (mikroblogowanie) ułatwia identyfikację nastrojów użytkowników. Wynika to przede wszystkim z prostych struktur gramatycznych oraz bezpośredniego wyrażania emocji często stosowanych w takich formatach. Poprzez kondensację złożonych myśli w krótkie stwierdzenia, mikroblogowanie zwiększa czytelność, przy zachowaniu zwięzłej ekspresji emocjonalnej. Platformy mikroblogowe stały się powszechnie używanym narzędziem komunikacji wśród użytkowników Internetu, a miliony osób codziennie dzielą się swoimi opiniami na temat różnych aspektów życia, co daje ogromne, stale aktualizowane repozytoria treści. Platformy te stanowią zatem cenne źródło danych do eksploracji analizy nastrojów (Pak & Paroubek, 2010). Należy jednak pamiętać o wyzwaniach etycznych związanych z analizą danych pochodzących z mikroblogów, w tym o ochronie prywatności użytkowników, a także odpowiedzialnym wykorzystywaniu wyników badań.

Warto zauważyć, jak wiele dyscyplin zauważyło rzeczywiste zastosowania i wykorzystuje wnioskowanie z treści komentarzy. Wiele marek nie tylko przeniosło swoje sklepy do sieci Internet, ale także stara się tworzyć sieć społeczną wokół swoich produktów na portalach społecznościowych. Ten nowy kanał informacyjny, zapewnia nieustanny kontakt z nabywcą, aby móc informować go o nowościach, stworzyć więź, ale przede wszystkich umożliwić zbieranie informacji zwrotnej, która jest niezbędna do udoskonalenia usługi oraz zrozumienia nowych potrzeb. Na przykład marka Coca-Cola stworzyła kampanię reklamową "#ShareaCoke". Celem kampanii było stworzenie bardziej osobistej i angażującej więzi między konsumentami, a marką Coca-Cola poprzez zastąpienie jej kultowego logo na butelkach, a takżę puszkach popularnymi indywidualnymi imionami (thebrandhopper.com, 2024). Według Tima Grimsa (2013), z perspektywy internetowej społeczność Coca-Coli na Facebooku przez tę promocję ilość użytkowników na grupie dyskusyjnej w Anglii wzrosła o 3,5%, a globalnie o 6,8%. Dzięki analizie sentymentu wpisów generowanych przez użytkowników, firma mogła klasyfikować publikowane treści jako pozytywne, negatywne lub neutralne. Dzięki tej analizie Coca-Cola zauważyła, że reklama zmieniła nastawienie konsumentów. Kampania przyniosła 18,3 miliona wrażeń medialnych, przy jednoczesnym znacznym poprawieniu postrzegania marki wśród młodych ludzi (marketingmag.com.au, 2024).

Warto jednak spojrzeć na analizę sentymentu szerzej - mimo swojej użyteczności napotyka ona na wyzwania związane z interpretacją języka naturalnego (tabela 1), takie jak: rozpoznawanie ironii, sarkazmu oraz różnorodności kontekstów kulturowych (Tsytsarau & Palpanas, 2012). Ponadto, dynamiczna natura języka, wraz z ciągłym rozwojem slangu i nowych form ekspresji w mediach społecznościowych, wymaga od systemów analizy sentymentu ciągłego dostosowywania się i ulepszania.

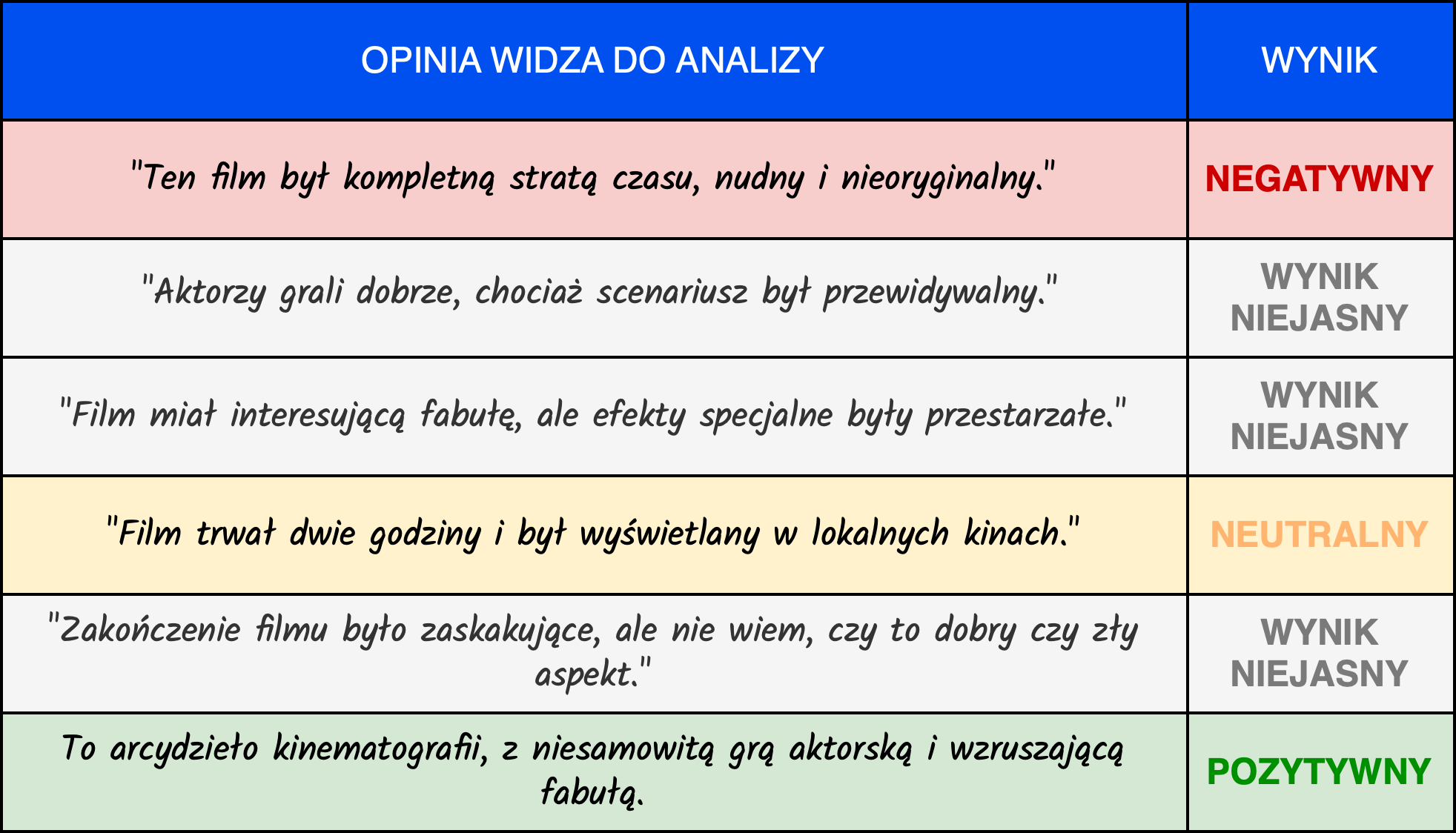


Tabela 1. Złożoność oraz niejednoznaczność sentymentu w przekazach tekstowych. Źródło: opracowanie własne

Analiza sentymentu, która początkowo opierała się na manualnych procesach, zyskała na znaczeniu dzięki dynamicznym postępom w dziedzinie informatyki. W przeszłości analiza ta była czasochłonna i ograniczona, jednak rozwój nowoczesnych komputerów, sprzętu oraz oprogramowania znacząco zwiększył jej efektywność i popularność. Obecnie, zaawansowane algorytmy wraz z potężną mocą obliczeniową umożliwiają głęboką analizę emocji wyrażanych w tekstach na niespotykaną wcześniej skalę. Jako integralna część uczenia maszynowego, przetwarzanie języka naturalnego (NLP) doświadczyło znaczących innowacji i usprawnień, stając się bardziej zaawansowaną technologią. Przez wiele lat była ona blokowana przez brak wystarczającej ilości danych oraz niewystarczające rozwiązania sprzętowe, które nie były w stanie wykonywać potrzebnej ilości obliczeń lub trwałyby one nieakceptowanie długo. Początkowe eksploracje w tym obszarze koncentrowały się na elementarnych rozwiązaniach, bazujących na leksykonach zawierających słowa oznaczone emocjami. Jednakże, to właśnie szybki rozwój w dziedzinie algorytmiki i adaptacyjnych modeli uczenia maszynowego, w tym uczenia głębokiego (ang. deep learning), spowodował, że analiza sentymentu przekształciła się w wysoce precyzyjne narzędzie, zdolne do dokładnych i wielowymiarowych analiz emocjonalnych zawartości tekstowej (rysunek 3). Zdolność do precyzyjnego rozpoznawania oraz interpretacji złożonych wzorców zachowań w obecnym trendzie technologicznym, gdzie dane stanowią kluczowy zasób, otwiera przed przedsiębiorstwami nowe horyzonty strategiczne, a przed naukowcami wiele obszarów eksploracji. Inteligentna analiza sentymentu, wspierana przez algorytmy uczenia maszynowego, pozwala na przetwarzanie ogromnych zbiorów danych z różnorodnych źródeł, takich jak media społecznościowe, fora internetowe, czy opinie klientów, dostarczając w ten sposób realnych i mierzalnych wskazówek dla decyzji biznesowych.

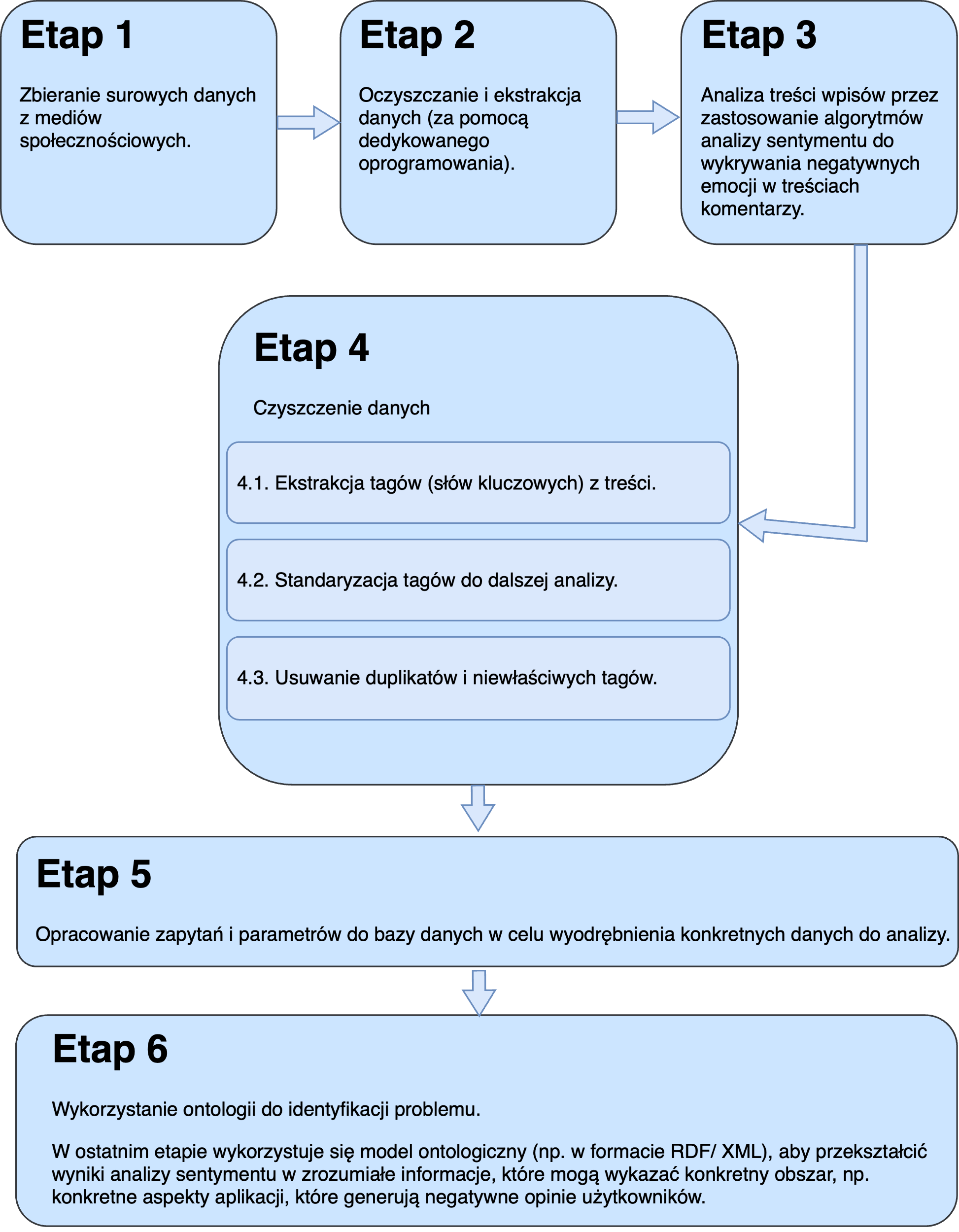
A screenshot of a computer

Description automatically generated

Rysunek 3. Porównanie głębokiego uczenia z uczeniem maszynowym. Źródło: opracowanie własne

Technika analizy sentymentu, przetwarzająca złożone dane tekstowe, jest kluczowa w wykrywaniu subtelnych sygnałów, które mogą być wykorzystane do optymalizacji oferowanych produktów i usług. Rozważmy sektor e-commerce. Badania Pang i Lee (2008) pokazały, jak analiza treści recenzji produktów dostarcza informacji wykraczających poza ogólne tendencje, wskazując na specyficzne cechy produktów cenione przez użytkowników. To umożliwia firmom dostosowanie oferty do precyzyjnie zidentyfikowanych oczekiwań rynku. W branży usługowej, jak wykazało badanie Valdivii, Luzon i Herry (2017), analiza recenzji na platformie *TripAdvisor* ujawnia niespójności między ocenami użytkowników, a treścią ich recenzji, co ma bezpośredni wpływ na zarządzanie reputacją i doskonalenie usług. Zespół badał, czy oceny numeryczne (tzw. „gwiazdki”) są zgodne z sentymentem wyrażonym w opinii. Badacze użyli do tego celu różnych metod analizy sentymentu, w tym opartych na leksykonach i uczeniu maszynowym. Celem było zrozumienie, jak niespójności wpływają na percepcję użytkowników oraz zarządzanie reputacją firm. Praca ta podkreśla, jak ważne jest nie tylko analizowanie ocen numerycznych, ale także głębokie zrozumienie treści tekstowych, aby uzyskać pełniejszy obraz odczuć klientów. Ta perspektywa dodaje nowego wymiaru, pokazując, że analiza sentymentu w e-commerce i branży usługowej wymaga uwzględnienia zarówno liczbowej reprezentacji oceny, jak i subtelności myśli wyrażonej w piśmie. W badaniu stwierdzono, że zgodność między pisemnymi recenzjami użytkowników, a ich ocenami numerycznymi jest niska. Zauważono, że negatywne recenzje często zawierają znaczącą liczbę pozytywnych słów. W badaniu odniesiono się do sytuacji, gdy w negatywnie klasyfikowanych krytykach pojawiają się wyrażenia, które trudno przypisać do jednej emocji, dobrym przykładem tutaj może być słowo: „*niedobrze*”. Zauważono, że mimo negatywnej oceny całej wypowiedzi, pozytywny wydźwięk był widoczny - średni procent występowania słowa „*dobry*” w tych recenzjach wynosi ok. 26%. Dla porównania skali, w recenzjach, które otrzymały wyższe oceny gwiazdkowe, procent ten wynosił ok. 39%. To badanie podkreśla złożoność analizy sentymentu w recenzjach online i wskazuje na potrzebę bardziej zaawansowanych technik analizy, aby lepiej zrozumieć i interpretować opinię. Rozumienie złożoności jak i niespójności w opinii klientów jest niebywale istotne w dążeniu do dostosowywania oferty w oparciu o rzetelne dane z recenzji.

Natomiast badanie przeprowadzone przez Thakora i Sasi (2015) prezentuje innowacyjne podejście do analizy sentymentu opartej na ontologii, które koncentruje się na wykrywaniu negatywnych sentymentów w treściach mediów społecznościowych. Autorzy skupili się na analizie krótkich wypowiedzi użytkowników na portalu Twitter (nowa nazwa marki brzmi: „X”), tzw. „tweet’ów” dotyczących firm świadczących usługi doręczeniowe, takich jak United States Postal Service, Royal Mail z Wielkiej Brytanii i Canada Post analizując kwestie takie jak opóźnienia w dostawach zgubionych paczek czy problem z obsługą klienta. Wykorzystując kombinację technologii do ekstrakcji danych z portalu internetowego X, oczyszczenia pobranych danych, analizy subiektywnej, budowy modelu ontologii oraz analizy sentymentu, naukowcy byli w stanie identyfikować i przetwarzać negatywne sentymenty związane z komentarzami (Thakor & Sasi, 2015). Jak widać na rysunku 4 w procesie budowy modelu, dane zostały oczyszczone z zagłuszających informacji, takich jak hiperłącza czy znaki specjalne, ponieważ mogłyby one utrudniać interpretację.



Rysunek 4. Przykładowy proces automatycznej analizy sentymentu dla treści pochodzących z mediów społecznościowych. Źródło: opracowanie własne

Wykorzystano skrypty napisane w języku programowania Python do automatycznego pozyskania i oczyszczenia treści, co było kluczowe w dalszym procesie badawczym. Następnie zastosowano metody przetwarzania języka naturalnego (NLP), dzięki którym można było zidentyfikować i oznaczyć kluczowe elementy tekstu, takie jak rzeczowniki i czasowniki. W tym modelu, klasy, obiekty i właściwości tych obiektów zostały określone odpowiednio jako encje. Korzystając z paradygmatu programowania obiektowego rzeczowniki zdefiniowano jako obiekty, a czasowniki jako właściwości tych obiektów w kontekście ontologii.

Jako następność firmy oferujące usługi kurierskie mogą wykorzystać opracowane rozwiązanie, wraz z syntezą wyników do podjęcia w odpowiedzi działań korygujących na zidentyfikowane problemy oraz do formowania zautomatyzowanej odpowiedzi w formie online na zgłoszone aspekty, co było decydującym krokiem w administrowaniu relacjami z odbiorcą oraz jakością oferowanych usług. Praca ta podkreśla jak istotne jest dogłębne, wielopoziomowe badanie każdej z grup: pozytywne, negatywne, neutralne.

W innej pracy "Sentiment Strength Detection in Short Informal Text" z grudnia 2010 roku autorstwa Mike'a Thelwalla, Kevana Buckleya, Georgiosa Paltoglou, Di Cai i Arvida Kappasa wskazano ponownie analizę sentymentu jako narzędzie do wykrywania krótkich, nieformalnych tekstów, takich jak omawiane wcześniej komentarze w portalach społecznościowych. Praca ta opiera się o badanie przeprowadzone przez zespół naukowców, którzy zwrócili uwagę na fakt, że większość istniejących algorytmów analizy sentymentu była zorientowana na identyfikacji opinii na temat produktów, a nie na wzorcach zachowań użytkowników. Podjęli się, dlatego, zadania częściowego wypełnienia tej luki poprzez stworzenie nowego algorytmu SentiStrength, który miał za zadanie ekstrahować siłę sentymentu z nieformalnych tekstów w języku angielskim, wykorzystując nowe metody do wydobycia de facto gramatyki i stylów pisowni. SentiStrength został zastosowany do analizy komentarzy na portalu internetowym MySpace i z wykorzystaniem tabeli poszukiwania siły sentymentu terminów zoptymalizowanej przez uczenie maszynowe, był w stanie przewidzieć pozytywne emocje z dokładnością 60% i negatywne emocje z dokładnością 73%, obie na skalach siły od 1 do 5. Podejście to było nowatorskie, ponieważ pozwalało na jednoczesne ocenienie siły sentymentu, co było nietradycyjne w porównaniu do zwyczajowych metod analizy sentymentu, które często koncentrowały się tylko na pozytywnych lub negatywnych kategoriach, a nie na wachlarzu emocji wrażanym przez ludzi. Przykład tej pracy pokazuje, że jak wiele płaszczyzn należy uwzględnić, aby poprawnie zaadresować przekaz wypowiedzi i emocje mu towarzyszące.

Centralnym punktem analizy sentymentu jest ukazanie wielorakich przekształceń danych tekstowych w cenne wskazówki, które firmy mogą wykorzystać do przewidywania trendów, personalizacji oferty i kształtowania strategii. Pizzerie stanowią istotny segment rynku gastronomicznego w USA, którego wartość w 2023 r. szacuje się na około 21,89 miliarda dolarów (Claight Corp. expertmarketresearch.com, 2024). W 2023 roku, około 40% sprzedaży w sieciach pizzerii w USA było powiązane z promocjami, czy też rabatami uzyskiwanymi głównie za pośrednictwem mediów społecznościowych (Ed Currington. thedigitalrestaurant.com, 2024). Dla porównania dane sprzed dziesięciu lat wskazują, że tylko około 5-10% sprzedaży w branży restauracyjnej mogło być bezpośrednio związane z promocjami w mediach społecznościowych (Andzulis et al., 2012). To pokazuje jak duża grupa odbiorców jest poddawana dyskusji oraz zachodzącym zmianom. Wzrost znaczenia mediów społecznościowych jako narzędzia marketingowego jest nie tylko wynikiem ich szerokiego zasięgu, ale także możliwości interakcji w czasie rzeczywistym z klientami, co pozwala na szybkie reagowanie na ich potrzeby. W erze cyfrowej, gdzie dane są nowym surowcem, efektywne wykorzystanie tej techniki staje się kluczem do sukcesu. Dlatego też, ta praca skupi się na analizie sentymentu z danych z mediów społecznościowych, blogów, aby w kolejnych krokach umożliwić pełniejsze zrozumienie nadawcy wiadomości, a przez to dokonać dochodowych decyzji biznesowych.

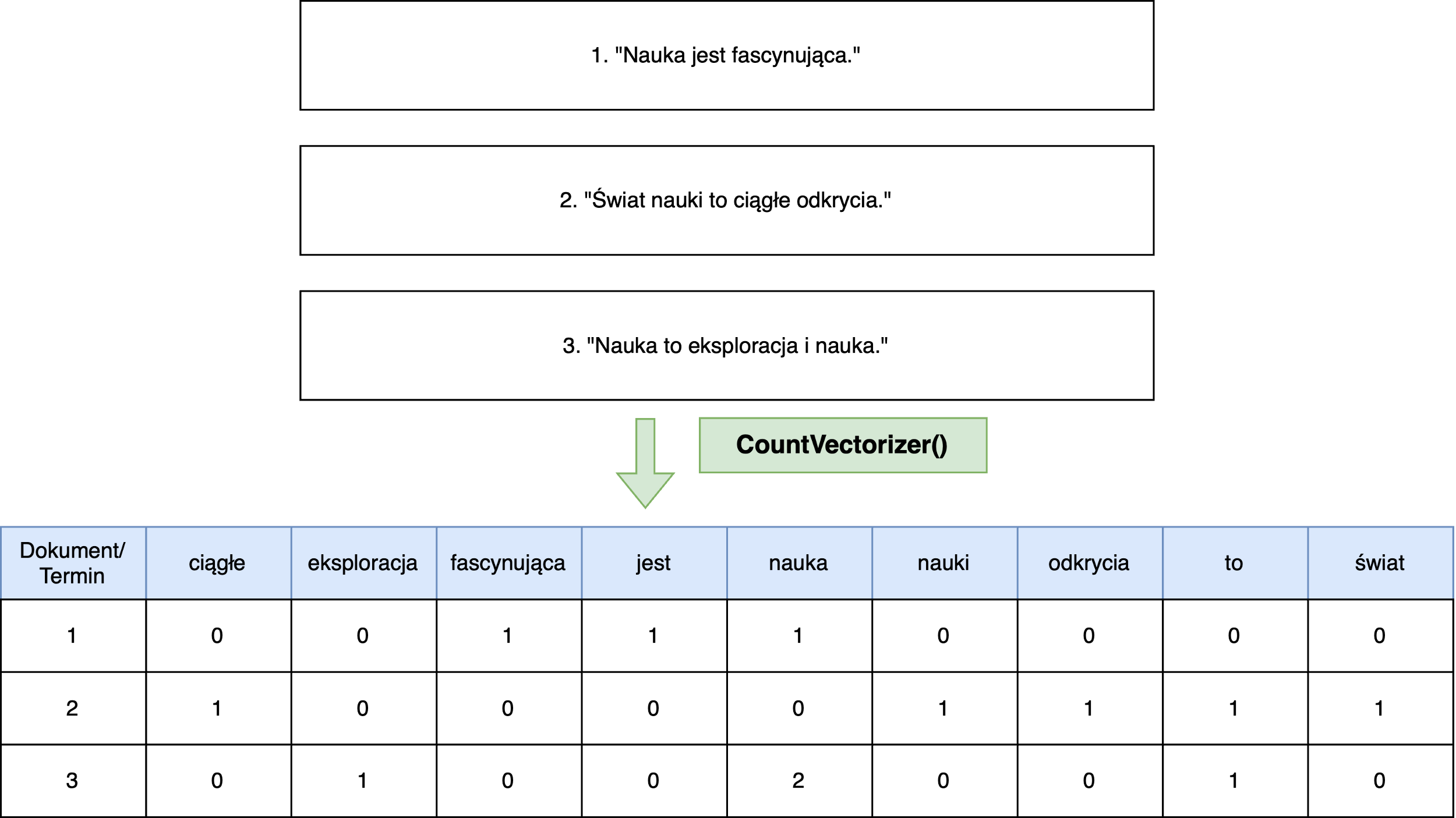
## 1.2 Przegląd narzędzi i technologii w Analizie Sentymentu

W analizie sentymentu, kluczową rolę odgrywają zaawansowane narzędzia i technologie, które umożliwiają precyzyjne przetwarzanie i analizę danych tekstowych. Wśród tych narzędzi wyróżnić można Regex, CountVectorizer i Natural Language Toolkit (w skrócie NLTK), które są niezbędne w arsenale specjalistów zajmujących się przetwarzaniem języka naturalnego.

Regex, czyli wyrażenia regularne, są potężnym narzędziem do wyszukiwania i manipulowania tekstem, pozwalającym na identyfikację specyficznych wzorców, takich jak słowa kluczowe czy frazy. Wyrażenia regularne oferują ograniczony, ale potężny *metalanguage* do opisywania wszelkiego rodzaju formatów, wyszukiwania słów, ekstrakcji informacji i wielu innych operacji. Wyrażenia regularne pojawiły się w kontekście teorii języków formalnych i teorii automatów, a ich podstawowe zastosowanie to mechanizm dopasowania wzorców tekstowych i analizy leksykalnej w kompilatorach (Erwig & Gopinath, 2012). W branży technologicznej, gdzie konieczna jest analiza logów systemowych czy kodów błędów, Regex umożliwia efektywne wyodrębnienie istotnych informacji z dużych zbiorów danych.

Term Frequency-Inverse Document Frequency (TF-IDF) to technika ważenia słów w dokumentach, która ocenia jak istotne jest określone słowo w kontekście całego korpusu tekstów. Pierwszy komponent to TF (ang. Term Frequency), który mierzy częstość wystąpień słowa w dokumencie, a IDF (ang. Inverse Document Frequency) ocenia rzadkość słowa w całym korpusie. Kombinacja tych miar pozwala na lepsze wyróżnienie ważnych słów w dokumentach, co jest użyteczne w zadaniach takich jak wyszukiwanie informacji i klasyfikacja tekstów (tajniki tej techniki są przybliżony w szczegółach w sekcji 2.3.1).

CountVectorizer, często stosowany z bezpłatnej biblioteki oprogramowania do języka Python scikit-learn, służy do konwersji kolekcji dokumentów tekstowych na macierz liczb całkowitych, reprezentujących częstotliwość występowania słów (rysunek 5). Jest to kluczowe w procesie Bag-of-Words, gdzie tekst jest przekształcany w zestaw jego składników, co jest nieocenione przy identyfikowaniu dominujących trendów i postaw w opinii klientów.



Rysunek 5. Przykłady użycia funkcji CountVectorizer dla różnych dokumentów. Źródło: opracowanie własne

Powyżej omówione techniki przetwarzania języka naturalnego pozwalają na podstawowe przygotowanie tekstu do dalszej analizy. Metody oparte na słownikach, uczeniu maszynowym oraz technikach głębokiego uczenia są fundamentem współczesnych systemów NLP.

Analiza słownikowa (ang. lexicon-based sentiment analysis) opiera się na uprzednio definiowanych słownikach, które zawierają listy słów wraz z przypisanymi wartościami sentymentu (np.: pozytywnymi i negatywnymi). Proces oceny ogólnego sentymentu polega na porównaniu słowa z analizowanego tekstu, z listą wyrazów w słowniku, a potem wartości sentymentu wszystkich słów są sumowane. Przykładami słowników używanych w tej metodzie są: SentiWordNet, VADER (opisany bliżej w sekcji 3.3) oraz AFINN. Zdecydowaną zaletą tego rozwiązania jest prostota implementacji, brak wymagań co do ilości danych treningowych, jednak przez to charakteryzuje się ono ograniczoną dokładnością, szczególnie w przypadku złożonych zdań czy idiomów.

Konkurencyjną dokładnością charakteryzują się natomiast metody oparte na uczeniu maszynowym. Proces przewidywania sentymentu w tym przypadku rozpoczyna się przetwarzaniem tekstu i zamianie na cechy przez TF-IDF, Bag-of-Words,czy n-gramy. Następnie klasyfikatory, takie jak regresja logistyczna, czy drzewa decyzyjne są trenowane, aby wytworzyć wzorce rozróżniające klasy. Metody te mają wiele zastosowań, jednak warto pamiętać, że do trenowania modeli wymagane są duże zbiory, wraz z dodanymi uprzednio oznaczeniami sentymentu.

Obecnie jako standard plasują się techniki głębokiego uczenia. Przykładami takich modeli są: recurrent neural networks, long short-term memory czy transformers (np. BERT, GPT). Charakteryzują się one zdolnością do uchwycenia skomplikowanych wzorców w danych tekstowych. Techniki głębokiego uczenia, mają unikalne struktury architektury, które pozwalają na osiąganie zadowalających wyników w analizie sentymentu, zazwyczaj lepsze niż tradycyjne modele ML. RNN mają ukryte stany, które przechowują informacje o poprzednich krokach, co w sekwencji pozwala im uchwycić kontekst. LSTM składają się z trzech typów bramek, które kontrolują przepływ informacji do komórek pamięci i z nich, co pozwala to na selektywne zapamiętywanie lub zapominanie informacji. Modele transformers skupiają się na częściach sekwencji z różną intensywnością, dzięki temu transformer może uchwycić relacje między słowami niezależnie od ich odległości w sekwencji. W analizie sentymentu narzędzia uczenia głębokiego są wykorzystywane w całym spektrum zapotrzebowań, od prostej klasyfikacji komentarzy na blogach, poprzez wirtualną obsługę klienta, po asystentów głosowych. Warto jednak pamiętać, że wymagają one dużej ilości danych oraz zasobów obliczeniowych do treningu.

Przetwarzanie języka naturalnego (NLP) w języku programowania Python stało się przez lata kluczowym narzędziem badań. Python, dzięki swojej prostocie i elastyczności, oferuje bogaty ekosystem z dokumentacjami, bibliotekami i społecznością. Wśród najpopularniejszych bibliotek NLP w tym języku programowania wyróżnia się NLTK, spaCy, TextBlob, Stanford NLP oraz Transformers (Hugging Face). Wszystkie te pakiety oferują unikalne funkcjonalności i dają możliwość wykorzystania w wielu dziedzinach: od naukowych, po komercyjne wdrożenia w biznesie.

NLTK to wszechstronna platforma do budowania programów w języku Python, do pracy z danymi języka naturalnego, oferująca łatwe w użyciu interfejsy do korpusów i zasobów leksykalnych, jak również narzędzia do klasyfikacji, tokenizacji, stemmingu, tagowania i parsowania (które zostaną bliżej opisane w kolejnym rozdziale). W analizie sentymentu, NLTK pozwala na szczegółową analizę języka naturalnego, co ma istotne znaczenie dla firm technologicznych w interpretacji danych klientów z mediów społecznościowych czy ogólnych treści zamieszczanych online.

Kolejną nowoczesną biblioteką NLP, znaną ze swojej szybkości i efektywności jest spaCy. Zapewnia ona optymalizację dla dużych zbiorów danych, a przez to jest stosowana w aplikacjach produkcyjnych. Narzędzia tam zaimplementowane, umożliwiają przetwarzanie tekstu, takie jak tokenizacja, lematyzacja, POS tagging, dependency parsing oraz named entity recognition (NER).

Jeżeli programista poszukuje prostszego rozwiązania, to może użyć TextBlob, które upraszcza wykonywania podstawowych zadań NLP (tokenizacja, przypisywanie części mowy, lematyzacja). Narzędzie to może znaleźć swoich odbiorców wśród początkujących lub osób prototypujących aplikację NLP, gdyż oferuje intuicyjny interfejs oraz łatwą integrację z innymi bibliotekami.

Programiści chcący programować w wielu językach mogą użyć Stanford NLP opracowane przez Stanford University. Oferuje zaawansowane funkcje NLP, w tym tokenizację, lematyzację, dependency parsing, NER oraz analizę sentymentu.

Jedną z najbardziej popularnych bibliotek obecnie jest Transformers rozwijana przez firmę Hugging Face. Oferuje ona łatwą implementacją i gotowe modele do wielu zadań przetwarzania języka naturalnego, bazujące na Transformerach, takich jak BERT, GPT-4, RoBERTa. Modele tam dostępne służą do klasyfikacji tekstu, analizy sentymentu, generacji tekstu i tłumaczenia maszynowego. Naukowcy używają tej biblioteki do eksperymentowania z nowymi technikami NLP, publikowania artykułów naukowych i rozwijania nowych modeli. Firmy technologiczne implementują udostępnione tam rozwiązania w swoich usługach, takich jak asystenci głosowi, systemy rekomendacji, narzędzia do analizy tekstu, automatyzacja obsługi klienta i wiele innych.

Wykorzystanie dostępnych narzędzi w analizie sentymentu otwiera przedsiębiorstwom technologicznym możliwość nie tylko efektywnego przetwarzania dokumentów, ale także przekształcania surowych danych w cenne dla biznesu uwagi. W dobie cyfryzacji, adaptacja stanowi niezbędny element w budowaniu przewagi konkurencyjnej na rynku.

W dziedzinie analizy sentymentu, znaczący postęp jest widoczny przez ilość bogatej literatury naukowej czy dostępnym, innowacyjnym rozwiązaniom technologicznym. Rozważając kluczowe prace i badania, nasuwa się obserwacja, że rozwój tej sfery jest ściśle powiązany z ulepszaniami algorytmów sztucznej inteligencji i sprzętu komputerowego.

Wśród pionierskich prac wyróżnia się te, które skupiają się na automatycznym rozpoznawaniu emocji w tekście, wykorzystując zaawansowane algorytmy uczenia maszynowego. Badania prowadzone przez Pang i Lee, ukazują, jak maszyny mogą efektywnie klasyfikować teksty na podstawie ich tonacji emocjonalnej. Oprócz tego, w literaturze naukowej pojawiają się prace koncentrujące się na specyficznych wyzwaniach związanych z analizą sentymentu, takich jak rozpoznawanie ironii czy sarkazmu, co jest szczególnie istotne w kontekście mediów społecznościowych. Badania te podkreślają potrzebę ciągłego doskonalenia modeli językowych, aby mogły one skutecznie interpretować coraz bardziej złożone i subtelne formy komunikacji ludzkiej.

Porównując istniejące rozwiązania warto zwrócić uwagę na rozwój narzędzi takich jak IBM Watson, Google Cloud Natural Language API czy Amazon Comprehend. Te systemy, zbudowane przez przodujące firmy technologiczne, wykorzystujące zaawansowane algorytmy i głębokie sieci neuronowe, umożliwiają nie tylko analizę sentymentu na szeroką skalę, ale również pozwalają na dokładne rozpoznawanie subtelnych niuansów emocjonalnych w tekście. Zastosowanie technologii przetwarzania strumieniowego, jak Apache Kafka, czy Apache Flink pozwala na analizę sentymentu w czasie rzeczywistym, co będzie kluczowe dla szybkiego reagowania, automatycznych alertów, rekomendacji. Integracja z IoT (ang. internet of things) pozwala na integrację z inteligentnymi urządzeniami, a to daje sposobność do monitorowania emocji użytkowników w świecie realnym. Co istotne, aplikacja tych rozwiązań w biznesie otwiera drogę do głębszego zrozumienia potrzeb klientów, umożliwiając firmom tworzenie bardziej spersonalizowanych strategii marketingowych i komunikacyjnych.

Analiza sentymentu rozwija się także wraz z zaawansowanymi technologiami, takimi jak modele transformacyjne, dzięki którym można dokonać lepszego zrozumienia kontekstu oraz multimodalnej analizy danych. Analiza multimodalna umożliwia integrację analizy tekstu z analizą obrazu, wideo czy dźwięku, dając kompleksowe podejście. Personalizacja i kontekstualizacja analizy sentymentu, uwzględniająca kontekst kulturowy, językowy, jest kluczowa w poprawnej klasyfikacji. W globalnie zarządzanej organizacji, kontekstualizacja, jak również wielojęzyczne modele dają możliwość monitorowania opinii klientów na całym świecie. Dzięki temu jedna firma, może lepiej rozumieć nastroje, potrzeby w różnych regionach, bez potrzeby tworzenia rozwiązania dedykowanego dla, np. jednego języka. Natomiast zastosowanie metod *Explainable AI* (w skrócie: XAI) zwiększa interpretowalność modeli, co umożliwia ich aplikację w dziedzinach takich jak prawo czy medycyna, ale zapewnia też wyjaśnialność procesu dla odbiorcy biznesowego.

Etyczne zastosowanie technologii i zgodność z regulacjami, jak GDPR (od ang. General Data Protection Regulatio), są kluczowe w analizie sentymentu, ponieważ wymuszą one transparentne przetwarzanie, z poszanowaniem praw i wdrożeniem zabezpieczeń. Krok ten umożliwi integrację analizy sentymentu z systemami CRM i ERP, aby w bezpieczny sposób automatyzować procesy biznesowe, usprawniając zarządzanie relacji z klientem, jak również maksymalizację personalizacji oferty.

Kluczowe dla przyszłości efektywnego przetwarzania danych są nowatorskie podejścia *edge computing* oraz *federated learning*, które wpływają na prywatność, kontrolę, przetwarzanie na urządzeniach końcowych czy trenowanie modeli bez centralizacji.

Podsumowując, przegląd literatury i istniejących rozwiązań w dziedzinie analizy sentymentu wskazuje na dynamiczny rozwój tej branży, który jest napędzany zarówno przez prowadzone badania akademickie, jak i rosnące potrzeby biznesu. Dla przedsiębiorstw, inwestycja w te technologie to krok ku głębszemu zrozumieniu rynku i większemu dostosowaniu strategii do potrzeb współczesnych konsumentów, co w konsekwencji prowadzi do zwiększenia konkurencyjności i efektywności produkcyjnej.

## 1.3 Podstawy Przetwarzania Języka Naturalnego

Analiza sentymentu jest częścią przetwarzania języka naturalnego (ang. Natural Language Processing, w skrócie: NLP), które skupia się na interakcjach między językiem ludzkim, a komputerami. Znajduje się na przecięciu informatyki, sztucznej inteligencji i lingwistyki obliczeniowej (Lopez & Kalita, 2017). Techniki NLP służą do wydobywania subiektywnych informacji z danych tekstowych, umożliwiając zrozumienie uczuć i opinii (Liu, 2012). NLP jest zbiorczym terminem odnoszącym się do automatycznego przetwarzania komputerowego języków ludzkich. Obejmuje to zarówno algorytmy, które przyjmują tekst wytworzony przez człowieka jako dane wejściowe, jak i algorytmy, które wytwarzają naturalnie wyglądający tekst jako dane wyjściowe (Goldberg, 2015). Ta interdyscyplinarna branża umożliwia systemom komputerowym nie tylko interpretację, ale także zrozumienie i odpowiedź na ludzki język, co otwiera nowe możliwości w analizie dużej ilości danych tekstowych.

Yue Kang wspólnie z innymi autorami w swoim artykule zatytułowanym: „Natural language processing (NLP) in management research: A literature review.” opisują badanie, które polegało na przeprowadzaniu treści artykułów z zakresu wiedzy o zarządzaniu, używając słów kluczowych „natural language processing”. Autorzy przeszukali 123 artykuły z dwudziestu czterech czołowych czasopism biznesowych zidentyfikowanych przez Teksański Uniwersytet w Dallas. W wyniku selekcji określono, że 50 artykułów - było nie związanych z tą techniką, ponieważ proponowały one NLP jako rozwiązanie przyszłościowe, odrzucały ją lub były elementem w biografii autora. Prace, które adresowały zagadnienia z NLP pochodziły z wielu dziedzin nauki (rysunek 6). Wykazane przez nich wyniki wskazują, że w kolejnych latach technika ta będzie wzrastać na znaczeniu w biznesie i zarządzaniu.

A graph with blue bars

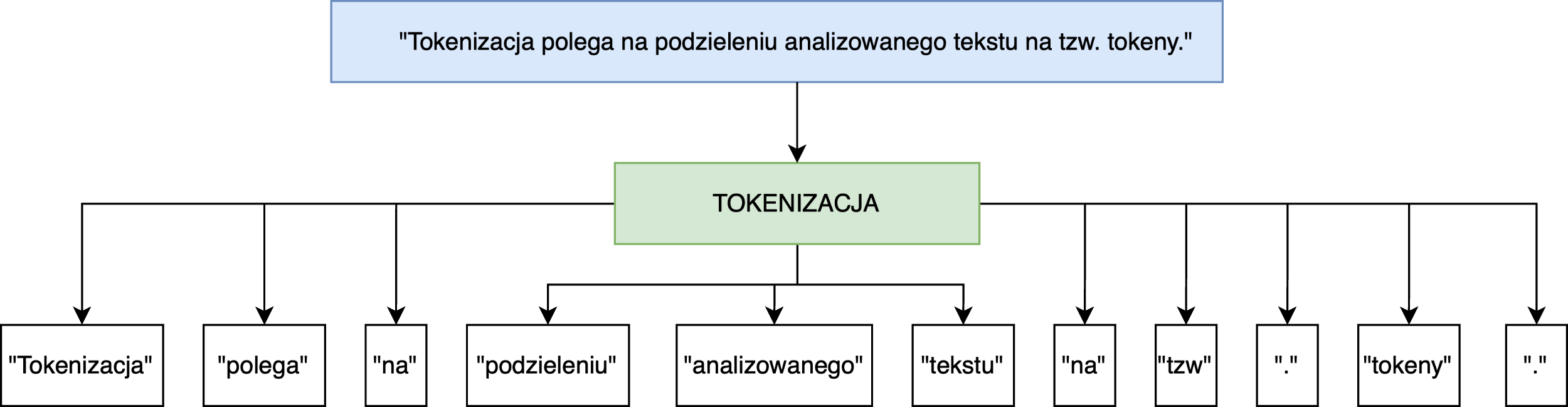
Description automatically generated

Rysunek 6. Rozkład publikacji dotyczących NLP w prestiżowych periodykach zarządczych. Źródło: opracowanie własne

Dzięki rozwojowi technik NLP przedsiębiorcy będą mogli efektywne wydobywać znaczenie, intencje i emocje z tekstu, co jest kluczowe w procesach decyzyjnych opartych na danych. W kontekście biznesowym NLP rewolucjonizuje sposób, w jaki firmy analizują informacje zawarte w tekście – zaczynając od opinii klientów na portalach społecznościowych przez recenzje produktów po komunikację wewnętrzną. Dzięki NLP, marki mogą automatycznie analizować duże zbiory danych tekstowych, co przekłada się na lepszą orientację na potrzeby klienta, szybszą reakcję na zmieniające się trendy rynkowe oraz bardziej celowane strategie marketingowe. Na przykład Nike korzysta z przetwarzania języka naturalnego do analizowania opinii klientów z różnych źródeł, w tym z mediów społecznościowych, recenzji produktów i interakcji z obsługą klienta. Analiza sentymentu jawi się jako potężne narzędzie dla użytkowników do wydobywania potrzebnych informacji, jak również do agregowania zbiorowych uczuć wynikających z recenzji. W ostatnich latach na pierwszy plan wysunęły się różne metody realizacji tego zadania (Devika, Sunitha, & Ganesh, 2016).

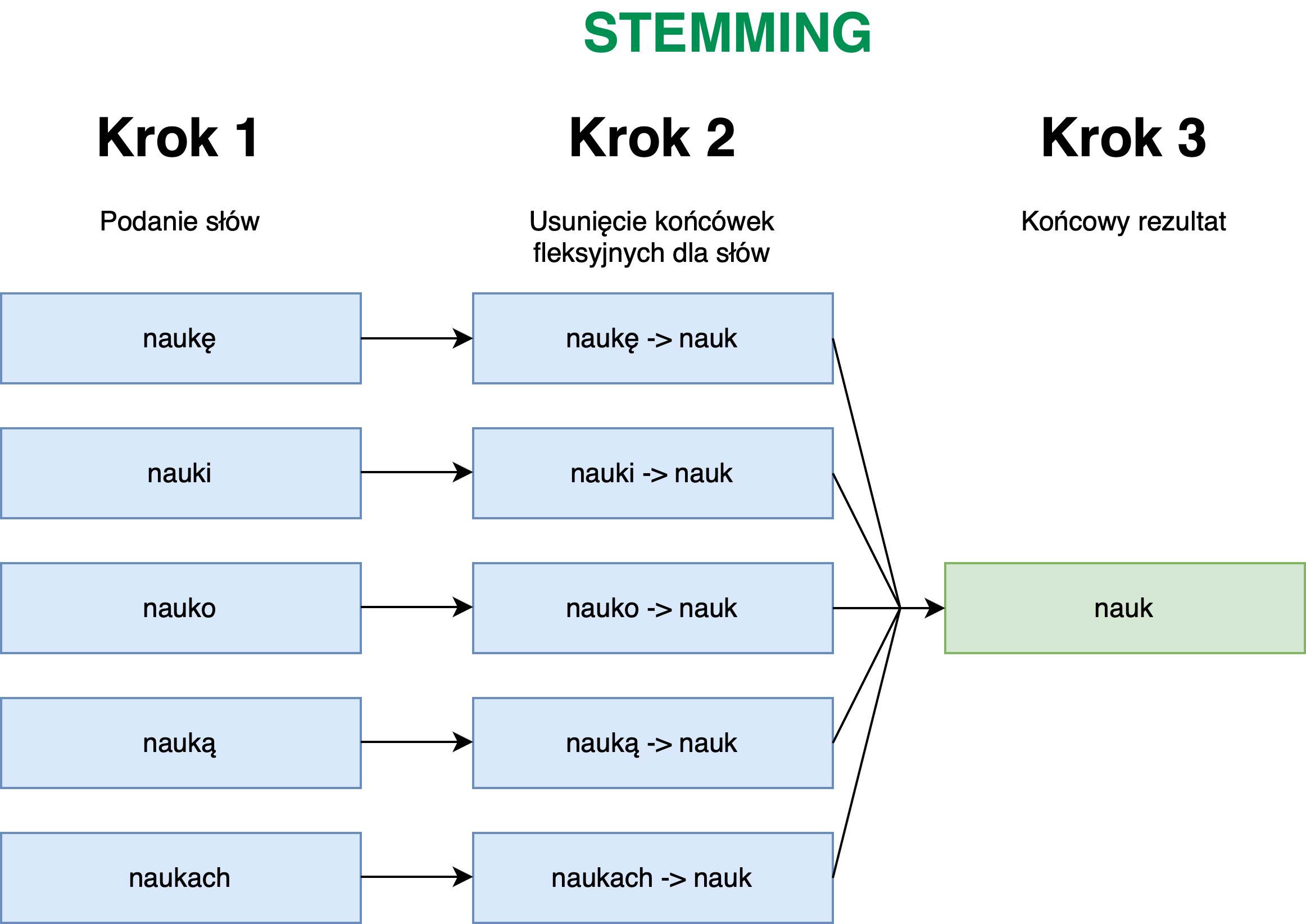
Technologie NLP, wykorzystując zaawansowane modele językowe, pozwalają na głębsze zrozumienie subtelności języka, co ma znaczący wpływ na jakość analizy sentymentu. Za pomocą technik takich jak tokenizacja, stemming i lematyzacja, systemy NLP potrafią przekształcać surowe dane tekstowe w strukturę, którą można łatwiej analizować i interpretować. To pozwala na wyłapywanie nie tylko ogólnych tendencji w wyrażanych opiniach, ale również na identyfikację bardziej złożonych wzorców i niuansów emocjonalnych, co jest nieocenione w tworzeniu strategii biznesowych opartych na danych. W ramach przetwarzania informacji pochodzących z opinii klientów, przedsiębiorstwa mogą zautomatyzować i usprawnić procesy analizy danych tekstowych, co pozwala na wydobywanie cennych wglądów i informacji strategicznych w znacznie szybszym czasie niż tradycyjne metody. W świecie biznesu, gdzie szybkość i precyzja decyzji mogą być decydujące, zastosowanie NLP w analizie sentymentu stanowi kluczowe narzędzie, umożliwiające firmom pozyskanie przewagi konkurencyjnej.

W ramach przetwarzania języka naturalnego (NLP), wykorzystuje się szereg technik, które mają kluczowe znaczenie dla analizy oraz interpretacji języka, szczególnie w branży technologicznej. Tokenizacja, czyli proces dzielenia tekstu na mniejsze jednostki - tokeny, zwykle – słowa (rysunek 7), jest niezbędna dla dalszego przetwarzania języka. Na przykład, w analizie danych z mediów społecznościowych, tokenizacja pozwala na efektywne rozpoznanie kluczowych słów i zwrotów, co jest kluczowe dla algorytmów śledzących trendy i nastroje użytkowników. Jednak warto zauważyć, że istnieje wiele pojęć dotyczących tego, co w NLP (przetwarzaniu języka naturalnego) jest uznawane za „token”. Różne pojęcia zależą od różnych celów (np. parsowania, MT) i często od różnego tła językowego. Aby dojść do definicji tokena, który jest zarówno lingwistycznie znaczący, jak i metodologicznie użyteczny należy zaadresować problem definicji pojęcia „słowa” z leksykograficznego punktu widzenia (Kit & Webster, 1992).



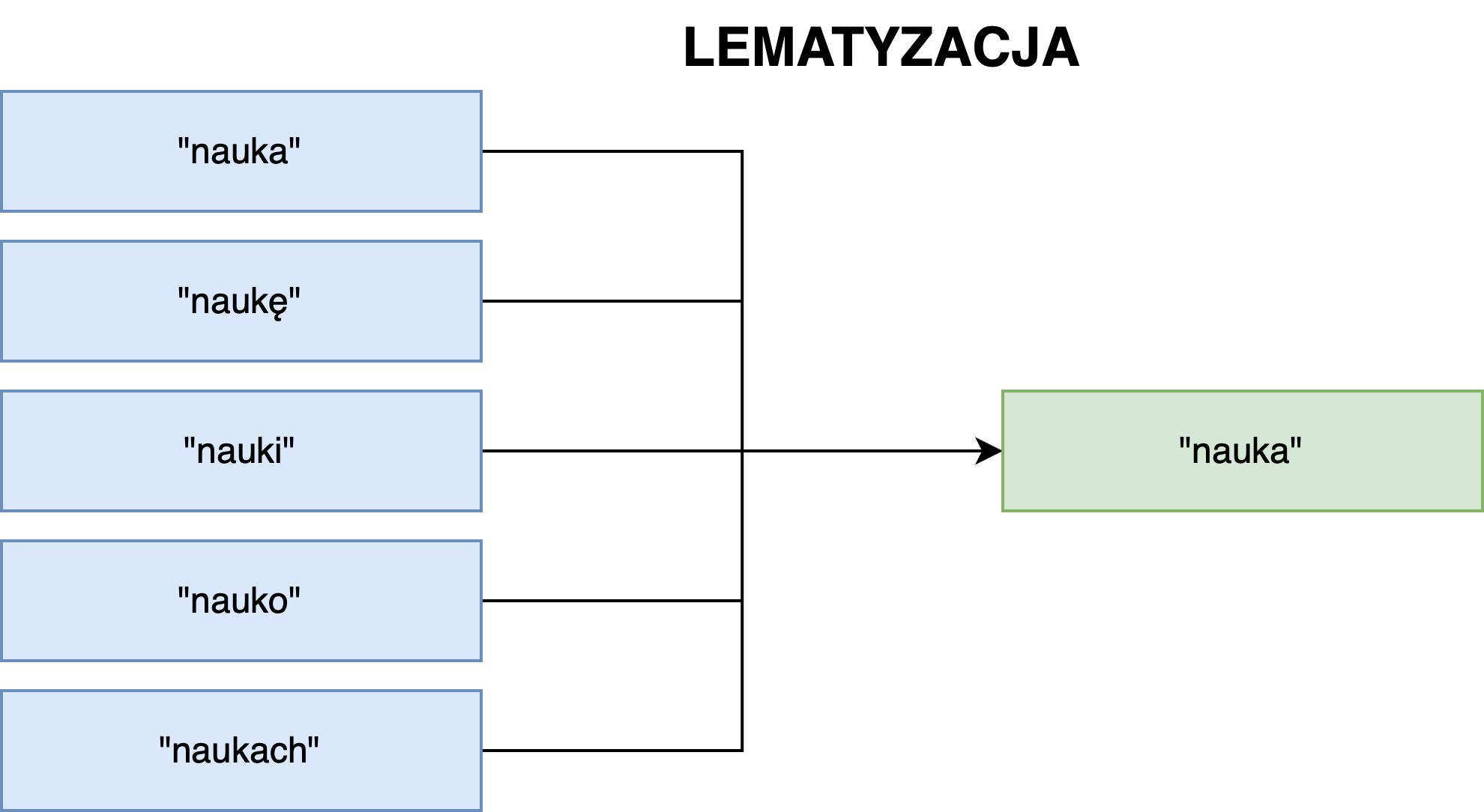
Rysunek 7. Przykład procesu tokenizacji. Źródło: opracowanie własne

Stemming to proces redukcji słów do ich podstawowych form, jest wykorzystywany do usprawnienia procesów wyszukiwania i klasyfikacji tekstu. W procesie stemmizacji 'rdzeń' słowa jest uzyskiwany poprzez zastosowanie zestawu reguł, ale bez uwzględniania części mowy (POS) czy kontekstu wystąpienia słowa (rysunek 8). Stemming jest często postrzegany jako narzędzie zwiększające metrykę recall, czyli skuteczności systemu wyszukiwania informacji. W kontekście stemmingu, który polega na redukcji słów do ich podstawowych form (np. "naukach" do "nauk"), recall odnosi się do zdolności systemu do identyfikowania dokumentów zawierających różne formy danego słowa. W językach o stosunkowo prostej morfologii wpływ stemmingu jest mniejszy, niż w językach o bardziej złożonej budowie morfologicznej (Jivani, 2011). Na przykład, w wyszukiwarkach internetowych, takich jak Google, stemming pozwala na zwiększenie trafności wyników wyszukiwania, dzięki czemu użytkownicy otrzymują bardziej precyzyjne informacje.



Rysunek 8. Przykłady stemmatyzacji dla różnych odmian słowa "nauka". Źródło: opracowanie własne

Lematyzacja, bardziej zaawansowana technika niż stemming, bierze pod uwagę kontekst i znaczenie słowa w zdaniu (rysunek 9), a następnie znalezieniu „lematu”, co jest niezwykle cenne w analizie semantycznej tekstu. Algorytm, który przekształca słowo na jego lingwistycznie poprawną formę podstawową, nazywany jest *lemmatizer*. W morfologii lemma kanoniczna forma to *lexeme*. *Lexeme* w tym kontekście odnosi się do zestawu wszystkich form o tym samym znaczeniu, a lemma to szczególna forma wybrana przez konwencję, aby reprezentować lexeme’m. (Jivani, 2011). W branży technologicznej, na przykład w asystentach głosowych typu Siri lub Alexa, lematyzacja pozwala na bardziej precyzyjne rozumienie zapytań użytkowników i dostarczanie odpowiedzi bardziej adekwatnych do ich intencji.



Rysunek 9. Przykłady lematyzacji dla różnych odmian słowa "nauka". Źródło: opracowanie własne

Zastosowanie tych technik NLP w branży technologicznej ma szerokie implikacje biznesowe. Umożliwiają one firmom technologicznym analizowanie ogromnych zbiorów danych tekstowych, co jest kluczowe w kontekście Big Data. W epoce cyfrowej, gdzie dane tekstowe są nieustannie generowane i przetwarzane, umiejętność efektywnego wykorzystania technik NLP staje się niekwestionowalna. Pozwala im to nie tylko na uzyskanie przewagi konkurencyjnej, ale także na podejmowanie świadomych i strategicznych decyzji biznesowych opartych na dogłębnej analizie dostępnych informacji.

Te techniki są fundamentem dla bardziej zaawansowanych operacji w NLP, umożliwiając efektywne przetwarzanie tekstu. Poprzez ich zastosowanie, systemy NLP mogą lepiej radzić sobie z różnorodnością i złożonością języka naturalnego, co przekłada się na dokładniejsze i bardziej wiarygodne wyniki. Przyszłość NLP jest ściśle związana z rozwojem sztucznej inteligencji. W miarę jak czytelność języka naturalnego się poprawia, komputery będą mogły uczyć się z informacji dostępnych online i stosować je w rzeczywistości. W połączeniu z generowaniem języka naturalnego, komputery będą coraz bardziej zdolne do odbierania i przekazywania użytecznych i wartościowych informacji lub danych (Chopra, Prashar, i Sain, 2013). Skomplikowane procesy w handlu (takie jak prognozowanie i podejmowanie decyzji podczas handlu akcjami), upraszczanie interakcji z klientami za pomocą wirtualnych asystentów na platformach handlowych (czyniąc interakcję bardziej przyjemną), analizowanie problemów obywateli w e-administracji, skuteczne zarządzanie operacjami w ochronie zdrowia (takie jak diagnozowanie, dostarczanie usług i zarządzanie dokumentacją) oraz ulepszanie podejść dydaktycznych w sektorze edukacji, to niektóre z korzyści, które mogą przynieść techniki NLP. Ponadto, integracja NLP z zaawansowanymi technologiami, takimi jak ML, AI i DL, może zapewnić jeszcze dokładniejsze wyniki odpowiadające na potrzeby użytkowników (Wu i Mircea, 2021).

## 1.4 Potencjalne wyzwania i ograniczenia

W analizie sentymentu, napotykamy na szereg wyzwań i ograniczeń, które stanowią przedmiot zainteresowania zarówno w kontekście technologicznym, jak i biznesowym. Jednym z głównych wyzwań jest skomplikowanie języka naturalnego, który charakteryzuje się nie tylko bogactwem ekspresji, ale także ironią, sarkazmem oraz zmiennymi kontekstami kulturowymi. Analiza sentymentu jest zadaniem subiektywnym i dlatego posiadane informacje mogą nadal cierpieć na uprzedzenia etykietowania, które dotykają zdecydowanej większości zbiorów danych (Muhammad et al., 2023). Te aspekty języka mogą prowadzić do ambiwalencji w interpretacji emocji, co staje się problemem szczególnie w automatycznej analizie tekstów pochodzących z mediów społecznościowych.

Dynamiczna ewolucja języka stanowi kolejne wyzwanie. Nieustannie dodawane są do słownika sformułowania, kreowane są nowe skróty, czy Emoji, a to wymaga od systemów NLP elastyczności i adaptacyjności. Stały rozwój algorytmów AI, które potrafią śledzić i dostosowywać się do tych zmian, jest kluczowy dla efektywnej klasyfikacji sentymentu.

Dodatkowo, przetwarzanie oraz analiza ogromnych ilości danych generowanych przez media społecznościowe wymaga zaawansowanej infrastruktury technologicznej oraz efektywnych metod transformowania danych. Wyzwanie to dotyczy zarządzania zapotrzebowaniem na energię elektryczną, budynkami oraz umiejętnością planowania zapisu danych w odpowiednich formatach, dedykowanych do potrzeb. Analiza sentymentu, zwłaszcza przy użyciu modeli głębokiego uczenia, może być intensywna obliczeniowo. Szkolenie czy samo uruchamianie modeli na dużych zbiorach danych wymaga znacznej mocy obliczeniowej, a także pamięci. Wraz ze wzrostem ilości danych, infrastruktura musi się odpowiednio skalować, a obejmuje to nie tylko pamięć masową, ale także zdolność do szybkiego przetwarzania i interpretacji wyników. Warto także pamiętać, że prowadzenie badań na dużą skalę, zużywa znaczną ilość energii elektrycznej, co zwiększa koszty operacyjne oraz wymaga podjęcia działań związanych z ochroną środowiska. Opracowanie bardziej energooszczędnych algorytmów i wykorzystanie zoptymalizowanego sprzętu, może pomóc złagodzić wysokie zużycie energii. Ponadto, nawet gdy infrastruktura jest gotowa, włączenie nowych komponentów do istniejących systemów biznesowych może być skomplikowane. Zapewnienie kompatybilności wraz z płynną integracją z innymi narzędziami jest niezbędne. Wdrażanie modeli w środowiskach produkcyjnych musi wymagać skalowalnych rozwiązań, zapewniających obsługę rzeczywistych danych dostarczających przydatnych informacji.

Przedsiębiorstwa działają w kontekstach, które są dynamiczne, podlegają ciągłym przeobrażeniom, składających się z aktorów takich jak konkurenci czy rząd, a wszystko to ma wpływ na to, jak organizacja może, a także powinna prowadzić biznes. To z kolei wywiera różne rodzaje presji na organizacjach ze względu na ich zdolność i skłonność do przyjęcia sztucznej inteligencji (Enholm, Papagiannidis, Mikalef, & Krogstie, 2021), aby przodować na rynku. Współcześni konsumenci oczekują spersonalizowanych doświadczeń, dokładnych odpowiedzi, a AI może usprawnić obsługę klienta, dostosować rekomendacje poprawiając interakcje z użytkownikami. Rozwiązania Data Science mogą zautomatyzować rutynowe zadania, usprawnić operacje oraz obniżyć koszty pracy, co ma kluczowe znaczenie w coraz bardziej konkurencyjnym kosztowo środowisku. Sztuczna inteligencja jest także obiecującym obszarem z perspektywy wykwalifikowanego pracownika, który szuka pracy z najnowocześniejszą technologią, innowacyjnymi projektami oraz zadaniami pozbawionymi powtarzalności, co pozwala się im skupić na zadaniach kreatywnych. W kontekście biznesowym, te wyzwania przekładają się na potrzebę inwestycji w nowoczesne technologie i edukację pracowników. Firmy, które chcą efektywnie wykorzystać analizę sentymentu, muszą podejmować świadome decyzje dotyczące wyboru instrumentów zdolnych do radzenia sobie z wymienionymi problemami. Wybór narzędzi jest bowiem ogromny – zaczynając od zaawansowanych modeli, takich jak konwolucyjne sieci neuronowe (CNN), rekurencyjne sieci neuronowe (RNN), a w szczególności sieci z długą pamięcią krótkotrwałą (LSTM), które mogą przechwytywać złożone wzorce w danych tekstowych. Firmy mogą też wykorzystać wstępnie wytrenowane modele językowe, takie jak GPT (Generative Pre-trained Transformer), który może generować tekst podobny do ludzkiego, rozumieć kontekst, dzięki czemu jest przydatny w zadaniach analizy nastrojów. Dostępne są również całe platformy do analizy tekstu, np. jako część pakietu Azure Cognitive Services, dostępny jest Microsoft Azure Text Analytic zapewniający analizę sentymentu, ekstrakcję fraz kluczowych, czy wykrywanie języka.

Podsumowując, chociaż analiza sentymentu oferuje obiecujące możliwości dla biznesu, wymaga ona jednocześnie skrupulatnego podejścia do istniejących wyzwań. Rozwiązanie tych niebanalnych problemów za pomocą przemyślanego i konsekwentnego podejścia jest kluczowe dla efektywnej implementacji oraz współpracy z NLP.

# 2 Uczenie Maszynowe w Analizie Sentymentu

Analiza sentymentu, znana również jako analiza emocjonalna tekstu, wykorzystuje zaawansowane algorytmy do oceny informacji zawartych w tekstach. Rozdział ten ma za zadanie teoretyczne wprowadzenie do algorytmów stosowanych w NLP, co pozwoli lepiej zrozumieć ich dalsze, praktyczne wykorzystanie w stworzonej aplikacji.

Uczenie maszynowe (ML od ang. Machine Learning) jest dziedziną łączącą sztuczną inteligencję i informatykę, skupia się na ulepszaniu i badaniu algorytmów systematycznych, które mają na celu uczyć się schematów na podstawie posiadanych danych oraz generalizować je. Algorytmy uczenia maszynowego są szeroko wykorzystywane do analizy sentymentu (Agarwal & Mittal, 2015). Uczenie maszynowe opisuje zdolność systemów do uczenia się na podstawie specyficznych dla problemu danych szkoleniowych, aby automatyzować proces budowania modeli analitycznych i rozwiązywać powiązane zadania (Janiesch, Zschech, & Heinrich, 2021). Freamwork uczenia maszynowego w kontekście ekstrakcji informacji, takim jak wydobywanie opinii, oferuje kilka zalet, w tym mniejszy wysiłek budowania w porównaniu do tworzenia symbolicznych reguł oraz probabilistyczne przypisanie, które jest cenne do oceny niepewności przypisania i stąd potrzeby obliczania kosztownych reprezentacji cech opierających się na przetwarzaniu języka naturalnego. (Boiy & Moens, 2009). Modele uczenia maszynowe są również bardziej adaptacyjne do specyficznych potrzeb branż i tematów. Ważnym aspektem jest również mniejsza subiektywność, która zapewnia większą przedmiotowość poprzez zastosowanie ustalonych modeli, reguł, niezależnie od osobistych przekonań lub oczekiwań. W analizie tekstu wykonywanej manualnie to od recenzenta zależy interpretacja danego fragmentu, co może prowadzić do niekonsekwencji wśród grupy osób. Dodatkowo modele ML dają wyjątkową możliwość doszkalania w kolejnych iteracjach, poprzez dodawanie nowych danych, ciągłe szkolenie, co wiąże się z możliwością udoskonalania o nowe wzorce, konteksty, bez wprowadzania ludzkiej subiektywności. Ostatecznie, prowadzi to do spójniejszych, powtarzalnych wyników, wszystkie te czynniki prowadzą do zwiększenia wiarygodności, weryfikowalność analizy danych mającego konsekwencje biznesowe.

Jest wiele różnych metod stosowanych analizy sentymentu (Rysunek 10). Wyróżnia się podejścia oparte na uczeniu maszynowym, leksykonie oraz metodach hybrydowych.

A diagram of a learning process

Description automatically generated

Rysunek 10. Klasyfikacja metod i technik analizy sentymentu. Źródło: Dang, N. C., Moreno-García, M. N., & De la Prieta, F. (2020). Sentiment Analysis Based on Deep Learning: A Comparative Study. Electronics, 9(3), 483.

Praca ta koncentruje się wokół podejścia wykorzystującego uczenie maszynowe ze względu na jego adaptacyjność i dokładność przewidywania, nawet dla zmieniających się danych wejściowych. Modele stworzone z podejściem uczenia maszynowego, a w jeszcze większym stopniu te zrealizowane w podgrupie uczenia głębokiego, wykazują zdolność do precyzyjniejszego wskazywania subtelnych detali językowych - jest to strategiczne podczas analizy emocjonalnych znaczeń w tekście pisanym. Kolejnym aspektem wartym poruszenia, jest kwestia, że uczenie maszynowe daje w tym momencie możliwość automatyzacji modeli w momencie, gdy nowa porcja danych jest dostępna. Podejście to ma też możliwość integracji różnych źródeł danych, struktur, formatów – co jest dodatkową korzyścią w multikulturowym środowisku globalnych społeczności internetowych. Kontekst analizy sentymentu wskazuje niejednokrotnie na potrzebę badania dużej ilości danych, gdzie uczenie maszynowe wykazuje możliwość przetwarzania, modelowania, prognozowania danych w skali, a osiągnięcie tego jest bardziej złożone dla metod znaczniej tradycyjnych.

## 2.1 Uczenie Nadzorowane

Uczenie nadzorowane (ang. Supervised Learning), czyli innymi słowy scenariusz nadzorowany charakteryzuje się pojęciem nauczyciela lub nadzorcy, którego głównym zadaniem jest dostarczenie informacji zwrotnej o błędach predykcji modelu (Bonaccorso, 2017). Jest to proces uczenia polegający na porównywaniu przewidywanych wartości z rzeczywistymi wartościami wyjściowymi (określonymi na początku) w celu oceny i korekty błędów.

Jest wiele zastosowań, gdzie uczenie nadzorowane umożliwia automatyczną oraz skuteczną analizę dużych ilości tekstów, dostarczając opiniotwórczych wiadomości dla biznesu. To uczenie jest stosowane w aplikacjach, gdzie dane historyczne przewidują prawdopodobne przyszłe wydarzenia. Istnieje wiele praktycznych zastosowań tego uczenia, na przykład aplikacja, która przewiduje gatunek irysa na podstawie zestawu pomiarów jego kwiatostanu. (Nasteski, 2017). Jako jedne z pierwszych zastosowań uczenia nadzorowanego było klasyfikowanie wiadomości elektronicznych, w celu zidentyfikowania spamu, ale także było używane do interpretacji pisma odręcznego ze zdjęć lub skanu, co miało służyć zautomatyzowaniu obróbki formularzy czy prawidłowemu rozpoznaniu kodu pocztowego na kopertach listów. Przez lata zauważono wyjątkowy potencjał tego rozwiązania i poszerzono użytkowania o inne dyscypliny, np.: medycynę, gdzie algorytmy pomagają w identyfikacji zmian nowotworowych w obrazach wyników badań medycznych. Jak wskazują artykuły z dziedziny, w opiece zdrowotnej, NLP przetwarza nieustrukturyzowane notatki kliniczne w celu wydobycia znaczących informacji, pomocnych w opiece nad pacjentem i badaniach medycznych (Savova et al., 2010). W finansach, NLP znalazło zastosowanie w przewidywaniu trendów na giełdzie papierów wartościowych, analizie ryzyka kredytowego czy automatyzacji doradztwa inwestycyjnego. Analiza języka naturalnego pomaga w wydobywaniu czynników ryzyka i specyficznych informacji z dużych ilości nieustrukturyzowanych danych finansowych. Pomaga w ten sposób w lepszej ocenie ryzyka i zarządzaniu nim (Kogan et al., 2009).

Ewolucja w dziedzinie przetwarzania języka naturalnego ciągle postępuje i odkrywane są kolejne przeznaczenia, m. in. w uczeniu głębokim, personalizacji informacji dla klienta czy rozpoznawaniu emocji. Warto zauważyć, w jak wielu obszarach analizy sentymentu można zaaplikować uczenie nadzorowane. W tej pracy wielokrotnie wspomniano o analizie emocji w mediach społecznościowych, ocenie recenzji, obsłudze klienta, ale jest też wiele kolejnych dyscyplin, których ważność tylko wzrasta, w tym także detekcja fałszywych wiadomości i dezinformacji. Klasyfikacja artykułów, wiadomośco pod kątem ich wiarygodności może obejmować analizę sentymentu w celu zidentyfikowania manipulacji czy fałszywych informacji. Szerokie rozprzestrzenianie się dezinformacji może mieć poważnie negatywny wpływ na jednostki, czy całe społeczeństwo - brak posiadania skalowalnych strategii weryfikacji faktów jest szczególnie niepokojący. Nic więc dziwnego, że ostatnie wysiłki badawcze są poświęcone nie tylko lepszemu zrozumieniu tego zjawiska, ale również automatyzacji wykrywania fałszywych informacji (Reis, Correia, Murai, Veloso, & Benevenuto, 2019).

### 2.1.1 Regresja Logistyczna

Regresja logistyczna umożliwia stosowanie ciągłych lub kategorycznych modeli prognozujących oraz charakteryzuje się możliwością dostosowania dla danych z wieloma zmiennymi zależnymi. Sprawia to, że jest ona szczególnie użyteczna do przewidywania wyniku, który jest kategoryczny (np. tak/nie). Zamiast przewidywać dokładne wartości, jak w regresji liniowej, regresja logistyczna przewiduje prawdopodobieństwo, że dany przypadek należy do określonej kategorii (LaValley, 2008).

Metoda ta jest stosowana do analizy sentymentu, ze względu na swoją prostotę i łatwą wyjaśnialność wyniku. Znajdzie ona swoje zastosowanie w przewidywaniu sentymentu w oparciu o różne cechy wyodrębnione z danych tekstowych takie jak: liczba słów, częstotliwość słów, n-gramy, a nawet bardziej wyrafinowane cechy, takie jak wyniki TF-IDF. Model regresji logistycznej ma za zadanie nauczyć się związków między tymi cechami, a etykietami sentymentu. Rozważmy firmę, która monitoruje opinie swoich klientów, aby zrozumieć jak odbierane są ich produkty. Ma ona zbiór danych oznaczonych recenzji (pozytywnych lub negatywnych) i chce zautomatyzować proces analizy nastrojów. Wybiera algorytm regresji logistycznej, który jest trenowany na tym zestawie. Model uczy się relacji między cechami recenzji, a etykietami nastrojów. Po przeszkoleniu model może przewidywać sentyment nowych, nieoznakowanych komentarzy. Firma może wykorzystać te przewidywania do oceny ogólnych nastrojów klientów, identyfikacji niedoskonałości i podejmowania świadomych decyzji dotyczących ulepszeń produkcji. Prostota algorytmu pozwala przedsiębiorstwu łatwo zinterpretować, które słowa lub frazy najbardziej wskazują na pozytywne lub negatywne nastroje, dostarczając praktycznych informacji dla zespołów analityków. Jednak słabością tego rozwiązania są wymogi: dużej ilości oznaczonych danych, liniowości zależności między zmiennymi, niezależność obserwacji, brak silnej korelacji w zmiennych niezależnych, równomierność rozkładu klas oraz ograniczenia w zdolności uchwycenia subtelnych niuansów.

Model regresji logistycznej używa funkcji logistycznej (sigmoidalnej) do przewidywania prawdopodobieństwa wystąpienia zdarzenia, że dana obserwacja należy do jednej z uprzednio określonych kategorii (rysunek 11).

A diagram of a function

Description automatically generated

Rysunek 11. Wykres funkcji sigmoidalnej. Źródło: Kleinbaum, D. G., & Klein, M. (2010). Logistic Regression: A Self-Learning Text (3rd ed.). Springer Science Business Media. ISBN: 978-1-4419-1741-6. DOI 10.1007/978-1-4419-1742-3.

Specyficzną charakterystyką wykresu funkcji jest jej kształt przypominający literę "S" oraz ograniczenie wynik do przedziału między 0, a 1, co sprawia, że ma silne zastosowanie do modelowania prawdopodobieństwa w kontekście klasyfikacji binarnej.

Wykres funkcji sigmoidalnej *f(z)*, gdzie oś pozioma reprezentuje zmienną *z*, która biegnie od do +, a oś pionowa przedstawia wartość funkcji *f(z)*, która mieści się w przedziale od 0 do 1.

Na wykresie (rysunek 11) można zauważyć kilka kluczowych punktów:

1. Punkt przecięcia wykresu z osią pionową w , gdzie . Oznacza to miejsce, w którym zmienna *z* jest równa 0. W tym punkcie wartość funkcji sigmoidalnej *f*(*z*) wynosi . Punkt ten jest istotny, ponieważ reprezentuje środek symetrii funkcji sigmoidalnej.
2. W miarę jak *z* zmierza do , wartość funkcji *f(z)* dąży do 0. Kiedy zmienna *z* staje się bardzo mała (ujemna i zbliża się do −∞), wartość funkcji *f*(*z*) dąży do 0. Oznacza to, że dla bardzo dużych ujemnych wartości *z*, funkcja sigmoidalna praktycznie osiąga wartość 0.
3. W miarę jak *z* zmierza do +, wartość funkcji *f(z)* dąży do 1. Kiedy zmienna *z* staje się bardzo duża (dodatnia i zbliża się do +∞), wartość funkcji *f*(*z*) dąży do 1. Oznacza to, że dla bardzo dużych dodatnich wartości *z*, funkcja sigmoidalna praktycznie osiąga wartość 1.

Objaśnienia w okręgach po lewej i prawej stronie wykresu opisują poniższe graniczne wartości funkcji:

(2.1)

- Dla .

- Dla .

Wartość funkcji w to , ponieważ .

Regresja logistyczna modeluje prawdopodobieństwo, że zmienna zależna Y = 1, na podstawie jednej lub więcej zmiennych niezależnych (oznaczonych zwyczajowo jako X). Wyżej wymienione prawdopodobieństwo jest przekształcone w logarytm szans (logit), który może być modelowany jako liniowa kombinacja zmiennych niezależnych. W regresji logistycznej, proces estymacji parametrów modelu (znanych również jako wagi) odbywa się za pomocą metody maksymalizacji funkcji wiarygodności. Co oznacza, że metoda ta dąży do znalezienia takiego zestawu wartości parametrów, aby prawdopodobieństwo wystąpienia zaobserwowanych wyników było jak największe. Algorytm rozpoczyna działanie z początkowym zestawem wag, które są zazwyczaj losowo wybrane lub inicjowane w inny standardowy sposób. Następnie, używając danych treningowych, algorytm regresji logistycznej iteracyjnie aktualizuje wyżej wymienione wagi. Każdy tego rodzaju powtarzalny krok ma na celu poprawę dokładności działania zbudowanego modelu w przewidywaniu wyników, czyli tym samym zmniejszyć błąd. Metodą najczęściej stosowaną do aktualizacji wag w regresji logistycznej jest gradientowy spadek (ang. gradient descent). Algorytm gradientowego spadku jest jedną z fundamentalny metod optymalizacji służąca do minimalizacji funkcji kosztu (lub funkcji straty) modelu, robi to za pomocą obliczania gradientu (kierunku i wielkości kroku) funkcji wiarygodności względem każdej, poszczególnej wagi, co w kolejnym kroku umożliwia aktualizację o taką wartość, która zwiększa wiarygodność przewidywań modelu. Aby dokonać decyzji o przynależności nowej obserwacji do klasy, porównuje się prawdopodobieństwo z ustalonym progiem, np. 0.5.W takiej sytuacji, jeżeli prawdopodobieństwo prognozowane jest większe niż próg, obserwacja jest klasyfikowana do jednej klasy, w przeciwnym razie do drugiej. Następnym krokiem, po dostosowaniu wag, jest ocena wydajności prognoz na zestawie danych testowych.

Regresja logistyczna ma swoje zastosowanie w wielu dziedzinach, takich jak medycyna (np. przewidywanie prawdopodobieństwa wystąpienia choroby), bankowość (np. ocena ryzyka kredytowego), marketing (np. przewidywanie odpowiedzi klienta na kampanię) i wiele innych.

Regresja logistyczna jest prosta w implementacji oraz interpretacji wyników, charakteryzuje się dobrą wydajnością w przypadkach, gdy zmienna zależna jest binarna oraz daje możliwość modelowania wpływu zmiennych ciągłych i kategorycznych. Jednak należy uwzględnić, że zakłada ona liniową zależność między logarytmem szans, a zmiennymi niezależnymi, co może nie być zawsze spełnione w rzeczywistych danych. Ma ona również swoje ograniczenia w modelowaniu zależności złożonych lub nieliniowych, co oznacza, że każdego razu, gdy następuje zmiana jednostkowa zmiennej niezależnej będzie mieć to swoje stałe odzwierciedlenie w logarytmie szans zdarzenia, a przy próbie uchwycenia złożonych wzorców w danych będzie to stanowiło wadę.

### 2.1.2 Metoda K - Najbliższych Sąsiadów

Intuicja leżąca u podstaw klasyfikacji metodą Najbliższego Sąsiada (w skrócie: k-NN od ang. k-Nearest Neighbour) jest dość prosta: przykłady są klasyfikowane na podstawie klasy ich najbliższych sąsiadów (Cunningham & Delany, 2021). Algorytm k-najbliższych sąsiadów może przewidywać zarówno atrybuty dyskretne (najczęściej występującą wartość wśród k najbliższych sąsiadów), jak i atrybuty ciągłe (średnią wśród k najbliższych sąsiadów) (Batista & Monard, 2002).

Algorytm KNN jest prostą, ale skuteczną metodą klasyfikacji w analizie sentymentu, która polega na znalezieniu najbliższych obserwacji nowego, nieznanego przykładu tekstu w zbiorze danych szkoleniowych i przypisaniu mu klasy na podstawie większości głosów. Załóżmy, że mamy recenzję o treści: "Restauracja ma wspaniałą obsługę, ale serwowane jedzenie jest zimne." Po przetworzeniu i wektoryzacji tej recenzji, model k-NN oblicza odległości do wszystkich recenzji w zbiorze danych. Znajduje najbliższych sąsiadów, np. 5 recenzji o podobnej treści, gdzie 4 są oznaczone jako pozytywne, a jedna jako neutralna. Na tej podstawie, model przypisuje nowej recenzji etykietę "pozytywna".

Prostota i intuicyjność modelu k-Najbliższych Sąsiadów sprawia, że jest to popularny algorytm w analizie tekstu, pomimo pewnych ograniczeń, takich jak konieczność przechowywania wszystkich danych szkoleniowych i obliczania odległości dla każdego nowego przykładu. Jako metryki do tego podejścia, stosuje się odległość euklidesową, Manhattan czy Minkowskiego. To podejście opiera się na założeniu, że podobne obserwacje (w sensie cech) będą należały do tej samej klasy. Jako przykład zastosowania, można przyjąć określenie sentymentu nowej opinii o restauracji jako pozytywny czy negatywny, używając metody k najbliższych sąsiadów, dla k = 3. Pragniemy analizować, jak bardzo nowa opinia przypomina trzy inne, wcześniej sklasyfikowane opinie, które w tej sytuacji mamy w danych i znamy ich sentyment. Po sprawdzeniu wybranych cech, wybrano trzy najbliższe opinii, które były pozytywne w dwóch przypadkach, a jedna z nich negatywna. Większość z nich była opiniami pozytywnymi, dlatego nową obserwację przypisujemy jako pozytywną. Można tu zauważyć alegorię z życia codziennego, gdy decydowanie o wyborze restauracji jest wykonywane na podstawie rekomendacji większości bliskich nam osób. Jeżeli osoby, które wykazują podobne gusta oceniają nową restaurację dobrze, to istnieje wysokie prawdopodobieństwo, że my też docenimy ich potrawy. Jest to zamysł często używany w rekomendacjach.

A diagram of numbers and points

Description automatically generated

Rysunek 12. Klasyfikacja metodą 3-najbliższych sąsiadów w dwuwymiarowej przestrzeni cech (Monthly\_Sal i Amount). Źródło: Cunningham, P., & Delany, S. J. (2020). k-Nearest Neighbour Classifiers 2nd Edition (with Python examples). arXiv:2004.04523v2 [cs.LG].

Powyższa grafika (rysunek 12) przedstawia wykres ilustrujący klasyfikację metodą 3-najbliższych sąsiadów (3-NN) w dwuwymiarowej przestrzeni cech, gdzie jedną z cech jest *Monthly\_Sal* (miesięczne wynagrodzenie), a drugą *Amount* (kwota). Na wykresie znajdują się dwa typy punktów oznaczone różnymi symbolami:

* Okręgi (o) reprezentujące jedną klasę.
* Krzyżyki (x) reprezentujące drugą klasę.

Dwa przykładowe punkty do klasyfikacji, oznaczone jako *q*1​ i *q*2​, są pokazane na wykresie:

* Punkt *q*1​ znajduje się bliżej grupy okręgów.
* Punkt *q*2​ znajduje się bliżej grupy krzyżyków.

Linie łączące punkty *q*1​ i *q*2​ z ich trzema najbliższymi sąsiadami ilustrują zasadę działania klasyfikatora 3-NN, który przypisuje nowy punkt do klasy większościowej spośród jego trzech najbliższych sąsiadów. Klasyfikator 3-NN klasyfikuje punkt *q*1​ jako należący do klasy okręgów (o), ponieważ większość z jego trzech najbliższych sąsiadów to okręgi. Analogicznie, punkt *q*2​ jest klasyfikowany jako należący do klasy krzyżyków (x).

Warto tu wspomnieć, że wybór parametru *k*, jest niebywale istotny - jest on kluczowy dla dokładności k-NN. Zbyt mała wartość parametru *k* może uczynić model wrażliwym na szum w danych, przez to zbyt dopasować się do danych treningowych (overfitting), podczas gdy zbyt duże *k* może prowadzić do zbyt ogólnej klasyfikacji, w takim przypadku model będzie niewystarczająco dopasowany (underfitting). Wybór optymalnej wartości *k* często wymaga eksperymentów i walidacji krzyżowej, a także wiedzy eksperckiej.

Dodatkowo brak konieczności trenowania modelu, czyni k-NN również leniwym algorytmem uczenia maszynowego (ang. lazy learning algorithm). Klasyfikatory leniwe nie trenują klasyfikatora do momentu przedstawienia próbki testowej (Garcia, Feldman, Gupta, Srivastava, et al., 2009). Algorytm ten wykazuje skuteczność w przypadkach, gdzie granica decyzyjna między klasami jest bardzo nieregularna nielinearna, k-NN nie robi założeń a priori o formie granicy decyzyjnej. W przeciwieństwie do wielu innych algorytmów uczenia maszynowego, które zakładają pewną prostotę lub regularność granic decyzyjnych (np. regresja logistyczna zakłada liniową separowalność klas), k-NN może dostosować się do bardzo nieregularnych i złożonych kształtów granic bez konieczności wprowadzania złożonych modyfikacji do modelu.

Wyraźną zaletą tego algorytmu jest prostota implementacji, przy zauważalnej intuicyjność działania, wynikającej z podobieństw płynących z życia codziennego.

### 2.1.3 Drzewo Decyzyjne

Drzewa decyzyjne (ang. Decision Tree) są graficzną metodą wspomagania procesu decyzyjnego (rysunek 13).

A diagram of a diagram

Description automatically generated

Rysunek 13. Drzewo decyzyjne. Źródło: Rozwadowska, A. (2022). Wspomaganie budowania modeli klasyfikacyjnych w przypadku niezbilansowanych klas w zbiorze danych.

Drzewa decyzyjne wywodzą swoje początki z okresu wczesnego rozwoju zapisu pisemnego. Ta historia ilustruje główną zaletę drzew: wyjątkowo łatwe w interpretacji wyniki, które mają intuicyjny, drzewopodobny wygląd, co z kolei wzmacnia zrozumienie i rozpowszechnianie wyników (De Ville, 2013). Transparentność i klarowność wnioskowania, prowadzą do łatwości w przedstawieniu go nawet dla osób, które nie są ekspertami w dziedzinie analizy danych.

Drzewa decyzyjne są cennym narzędziem w zestawie metod przetwarzania języka naturalnego, gdzie łatwość interpretacji, jakie cechy wpływają na wynik jest istotna. Nie wymagają normalizacji lub standaryzacji, co upraszcza przetwarzanie wstępne i można je trenować na danych ciągłych jak i kategorycznych. Drzewa decyzyjne można zastosować w analizie sentymentu opinii konsumenckich w e-commerce. Na przykład, komentarze zawierające frazy typu "świetny produkt" lub "polecam" będą klasyfikowane jako pozytywne, podczas gdy recenzje zawierające "strata pieniędzy" lub "odradzam" jako negatywne. Takie badania pomagają sklepom w identyfikacji popularnych produktów i tych, które wymagają wprowadzania zmian.

Mają one strukturę, które składają się z wierzchołków połączonych krawędziami. Wierzchołki, z których wychodzi co najmniej jedna krawędź, są nazywane węzłami, a pozostałe wierzchołki – liśćmi. To znaczy, że mają one jasno określoną hierarchię – od korzenia poprzez kolejne poziomy decyzji po liście (węzły końcowe bez dzieci). W każdym węźle sprawdzany jest określony warunek, zapytanie na temat danych dotyczący danej obserwacji i na jego podstawie wybierana jest jedna z gałęzi prowadząca do kolejnego wierzchołka. Punktem startowym działania algorytmu jest węzeł startowy, nazwany korzeniem, który jest nominalnym punktem wyjścia dla wszystkich decyzji. Każda gałąź odpowiada na zadane pytanie, prowadząc nad do kolejnych węzłów lub liści, które oznaczają koniec ścieżki oraz podają wyznaczony wynik. Proces decyzyjny w każdym etapie opiera się na ocenie obserwacji wejściowej według określonego kryterium np. czy wartość pewnej cechy jest większa od zadanej liczby. Na podstawie tej oceny, podejmowana jest decyzja o dalszej "ścieżce" – czyli którą gałąź wybrać, aby przejść do następnego węzła. Opisana procedura jest powtarzana, do momentu znalezienia się obserwacji w liściu, który zawiera klasyfikację oraz w ten sposób odpowiada ma pytanie, do jakiej klasy będzie należeć. Ten tok postępowania można postrzegać jako zachłanne przeszukiwanie przestrzeni wszystkich możliwych drzew decyzyjnych poprzez skanowanie instancji w danym węźle w celu określenia zysku z każdego podziału i wybrania pojedynczego podziału, który zapewnia największy zysk (Kotsiantis, 2013). Drzewa decyzyjne są faktycznie acyklicznymi grafami skierowanymi, ponieważ każda krawędzi ma określony kierunek i nie zawiera cykli. Model ten dzieli zbiór danych na mniejsze podzbiory na podstawie prostych zapytań, algorytm wybiera warunki na węzłach, które najefektywniej segregują dane, maksymalizując różnice między kategoriami danych w każdym liściu. To daje możliwość, aby model dokładnie predykował kategorię dla nowej obserwacji, bazując na posiadanych cechach z zestawu danych treningowych. Klasyfikacja nowego przypadku polega na przejściu od korzenia do liścia i przypisaniu do niej klasy zapisanej w danym liściu.

Drzewa decyzyjne są jednym z podstawowych narzędzi w analizie danych i uczeniu maszynowym. Są one łatwe do interpretacji, ponieważ strukturę drzewa można przedstawić graficznie, a my możemy podążać za gałęziami drzewa zgodnie ze zmiennymi wejściowymi, co wymaga mniej czasu na szkolenie (Sharma & Dey, 2012). Ich struktura, przypominająca drzewo z korzeniem, gałęziami i liśćmi, odzwierciedla sposób podejmowania decyzji poprzez serię binarnych (tak/nie) pytań dotyczących cech danych. Mają one umiejętność wykonywania predykcji na danych kategorycznych jak i numerycznych. Dodatkowym ich atutem jest brak wymagań w kwestii wstępnego przetwarzania danych, np. normalizacji, daje to unikatową możliwość dla badacza pracy na oryginalnych informacjach. Ta zaleta ma duże znaczenie w szczególności, gdy zachowanie oryginalnej skali i rozkładu ma kluczowe znaczenie dla przejrzystości modelu.

Pomimo wielu pozytywów warto spojrzeć na wady, ponieważ drzewa decyzyjne nie są ich pozbawione. Jedną z głównych jest ich skłonność do przeuczenia (overfitting), szczególnie gdy budowane drzewa są zbyt głębokie. Przeuczenie oznacza, że model zbyt dokładnie dopasowuje się do danych treningowych, włączając w to nawet szum, co może skutkować słabą generalizacją na nowych danych. Aby przeciwdziałać przeuczeniu, stosuje się różne techniki, takie jak przycinanie drzewa (ang. pruning), które polega na usuwaniu tych części drzewa, które nie przynoszą istotnej poprawy predykcji na danych walidacyjnych. Wykazywać one również niestabilność w niektórych sytuacjach, co oznacza, że małe zmiany w danych treningowych mogą prowadzić do znaczących różnic w strukturze drzewa. Ta cecha wynika z hierarchicznej natury drzewa, gdzie decyzje podjęte na wczesnych etapach mają duży wpływ na kształt całości. Ponadto, mogą mieć one trudności z uchwyceniem zależności liniowych między zmiennymi.

W przeciwieństwie do modeli liniowych, które eksplikują zależności między zmiennymi w sposób bezpośredni, drzewa decyzyjne mogą wymagać wielu podziałów w danych, aby zasymulować liniową zależność, co może prowadzić do nieefektywnego modelu. Pomimo określonych wyżej ograniczeń, drzewa decyzyjne stanowią podstawę dla zaawansowanych technik modelowania, takich jak lasy losowe (ang. Random Forests) czy boosting (np. Gradient Boosting). Metody te polegają na połączeniu wielu drzew decyzyjnych i skonstruowaniu zagregowanego systemu do predykcji odpowiedzi na zadane pytanie, aby uzyskać stabilniejszy, dokładny model, mniej podatny na małe zmiany. Lasy losowe działają poprzez budowę wielu drzew na losowo wybranych podzbiorach danych treningowych i cech, a ich predykcje są uśredniane, co zwiększa dokładność i odporność na przeuczenie. Boosting natomiast, polega na sekwencyjnym budowaniu drzew, gdzie każde kolejne drzewo próbuje skorygować błędy popełnione przez poprzednie, co prowadzi do stopniowej poprawy dokładności modelu.

2.2 Uczenie Głębokie

Głębokie uczenie (DL od ang. Deep Learning) to zastosowanie sztucznych sieci neuronowych (ANNs od ang. Artificial Neural Networks) do zadań uczenia się przy użyciu sieci o wielu warstwach (Zhang, Wang, & Liu, 2018). W tradycyjnych podejściach do uczenia maszynowego cechy są definiowane i ekstrahowane ręcznie lub za pomocą metod selekcji cech. Jednak w modelach głębokiego uczenia cechy są uczone i ekstrahowane automatycznie, co pozwala osiągnąć lepszą dokładność i wydajność (Dang, Moreno-García, & De la Prieta, 2020).

W analizie nastrojów modele głębokiego uczenia mogą być wykorzystywane do automatycznego uczenia się i wyodrębniania cech, a są one szczególnie skuteczne w scenariuszach, w których związek między tekstem, a sentymentem jest wysoce złożony. Na przykład firma zajmująca się analizą nastrojów komentarzy z mediów społecznościowych może wybrać model Transformer, taki jak BERT, dostrajając go do zbioru posiadanych danych z oznaczonym sentymentem. Model uczy się kojarzyć cechy tekstowe opinii z ich etykietami. Po przeszkoleniu pozwala to firmie monitorować w czasie rzeczywistym trendy nastrojów online, identyfikując pojawiające się kwestie lub pozytywne opinie. Zdolność systemu do zrozumienia kontekstu pomaga przedsiębiorstwu uzyskać więcej informacji, na przykład może odróżnić sarkazm od prawdziwego sentymentu, zapewniając dokładniejszą dedukcję. Informacje te mogą kształtować nowe strategie marketingowe, ulepszenie obsługi klienta i rozwój produktów. Głębokie uczenie oferuje potężne korzyści w analizie sentymentu, szczególnie w zakresie obsługi złożonych, wielowymiarowych danych tekstowych i wychwytywania niuansowych relacji między tekstem, a sentymentem. Pomimo wyzwań, takich jak wysokie wymagania obliczeniowe oraz problemy z interpretacją, doskonała wydajność przy jednoczesnej zdolności do automatycznego uczenia się funkcji sprawiają, że głębokie uczenie jest nieocenionym narzędziem w zaawansowanych aplikacjach do analizy nastrojów.

Modele głębokiego uczenia składają się z wielu warstw przetwarzających, gdzie każda warstwa odpowiedzialna jest za ekstrakcję i transformację cech na coraz wyższym poziomie abstrakcji. Struktura sieci neuronowej wykorzystywanej w analizie sentymentu zależy od wybranej architektury i złożoności zadania, które ma wykonać. W uproszczonej budowie sieć będzie składać się z trzech warstw. Pierwsza warstwa, zwana jest wejściową (ang. Input Layer), gdzie przyjmowane są dane wejściowe. W analizie sentymentu na tym etapie dzieje się tokenizacja tekstu na słowa, a następnie konwersja na wektory za pomocą technik takich jak one-hot encoding lub embedding słów (np. *Word2Vec, GloVe*). Embedding pozwala na reprezentowanie słów w przestrzeni wektorowej, ułatwiając modelowi zrozumienie kontekstowych oraz semantycznych relacji między nimi. Druga warstwa to warstwa ukryta (ang. Hidden Layers), w niej następuje przetwarzanie danych w celu ekstrakcji cech i wzorców. Można wyróżnić trzy typowe architektury dla warstwy ukrytej. Pierwsza to Rekurencyjne sieci neuronowe (RNN od ang. Recurrent Neural Network) – w tym zamyśle dane przetwarzane są sekwencyjnie, krok po kroku, przechowując informacje o wcześniejszych elementach w swoim "stanie ukrytym". Każdy neuron na tym etapie odbiera dane z dwóch źródeł: aktualnego elementu sekwencji (np. słowa w zdaniu) oraz stanu ukrytego poprzedniego kroku. Daje to RNN unikatową możliwość, aby "pamiętać" informacje, które pojawiły się w przeszłości i uwzględniać je przy przetwarzaniu nowych danych, co jest kluczowe dla zrozumienia kontekstu i znaczenia w sekwencjach tekstowych.

W typie sieci LSTM (ang. Long Short-Term Memory) i GRU (ang. Gated Recurrent Unit) pojawiają się dodatkowe mechanizmy kontrolujące przepływ informacji, takie jak bramki zapominania, wejściowe i wyjściowe. Narzędzia te pozwalają na skuteczniejsze zarządzanie pamięcią i przetwarzanie sekwencji danych, szczególnie przy uczeniu się długoterminowych zależności. W LSTM wykorzystuje się strukturę złożoną z bramek (zapominania, wejściowej i wyjściowej), regulujących jakie informacje powinny być przechowywane, aktualizowane lub odrzucane w każdym kroku czasowym. Te bramki umożliwiają modelowi zachowanie ważnych informacji przez dłuższy czas i skuteczne "zapominanie" danych uważanych za nieistotne. W GRU starano się uprościć strukturę, która buduje LSTM, dlatego połączono w jedno bramkę zapominania i wejściową, tworząc bramkę aktualizującą. Dodatkowo w GRU dokonano modyfikacji sposobu zarządzania stanem ukrytym. Takie podejście pozwala na redukcję liczby operacji i parametrów modelu, co często przekłada się na szybsze uczenie i równie dobre wyniki. Oba te podejścia, zarówno LSTM, jak i GRU wraz z zastosowaniem tych mechanizmów kontrolnych, umożliwiają sieci doskonalsze zarządzanie informacjami przez długie sekwencje, poprawiając zdolność do nauki wzorców, a także redukując problem zanikającego gradientu. Możliwości te poszerzyły wachlarz zadań, umożliwiając realizację złożonych zagadnień, które były trudne w wykonaniu przy użyciu wcześniejszych architektur RNN, czyni to je niezwykle wartościowymi w wielu zastosowaniach biznesowych.

Kolejnym typem sieci, który opiszę w tej pracy wykorzystuje konwolucyjne sieci neuronowe (CNN od ang. Convolutional Neural Network), których nazwa pochodzi od wykorzystywanych operacji na danych wejściowych, tzw. konwolucji. W tych zabiegach używane są zestawy filtrów (lub jąder), które w sposób automatyczny wykrywają istotne cechy w danych. Obecność określonych wzorców jest reprezentowana przez aplikowanie na różne fragmenty częścią danych wejściowych przez każdy filtr. W kontekście analizy tekstu, poszukiwane wzorce mogą dotyczyć specyficznych kombinacji słów czy też całych fraz wskazujących na określony sentyment. W warstwach ukrytych Transformerów (rysunek 14) następuje kalkulowanie znaczenia dla aktualnie przetwarzanych elementów, gdzie mechanizmy uwagi pozwalają każdemu elementowi sekwencji wejściowej (np. słowo w zdaniu) na ocenę czy powinien być on powiązany z innymi elementami tej sekwencji. To innowacyjne podejście, stanowi przełom w dziedzinie przetwarzania języka naturalnego (NLP), zostało one wprowadzone w pracy "Attention is All You Need" autorstwa Vaswani et al. W podanym dokumencie autorzy twierdzą, że transformer to pierwszy model transdukcji sekwencji oparty całkowicie na mechanizmach uwagi, zastępujący najczęściej używane w architekturach enkoder-dekoder warstwy rekurencyjne z wielogłowicową (ang. multi-headed) samo-uwagą (ang. self-attention) (Vaswani et al., 2017).

A diagram of a process

Description automatically generated

Rysunek 14. Model architektury transformera. Źródło: Vaswani, A., Jones, L., Shazeer, N., Parmar, N., Uszkoreit, J., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need.

Ostatnia warstwa to warstwa wyjściowa (ang. Output Layer), której zadaniem jest klasyfikacja sentymentu tekstu na podstawie przetworzonych i scharakteryzowanych danych z warstw ukrytych tak aby zostały przekształcone w konkretną decyzję predykcyjną. W analizie sentymentu głównym zadaniem warstwy wyjściowej będzie przekształcenie aktywacji (lub cech) otrzymanych z ostatniej warstwy ukrytej na konkretne kategorie sentymentu, takie jak pozytywny, negatywny, lub neutralny. Warstwa ta w swojej standardowej odsłonie składa się z zestawu neuronów, gdzie liczba neuronów odpowiada liczbie kategorii sentymentu do przewidzenia. Na przykład, w modelu rozróżniającym trzy kategorie sentymentu (pozytywny, negatywny, neutralny) warstwa ta będzie zawierać adekwatnie trzy neurony. Wybranie najbardziej prawdopodobnego sentymentu jako wyniku jest możliwe dzięki wykorzystaniu funkcji aktywacji, takiej jak „*softmax*”, na podstawie prawdopodobieństwa uzyskane z tej funkcji.

Dzięki zdolności do nauki reprezentacji na różnych poziomach abstrakcji, głębokie sieci neuronowe często osiągają lepszą generalizację na nowych danych, porównywalnie lub lepiej niż tradycyjne modele uczenia maszynowego.

A diagram of machine learning

Description automatically generated

Rysunek 15. Różnice między dwoma podejściami do klasyfikacji polarności sentymentu: uczenie maszynowe (góra) i głębokie uczenie (dół). Część mowy (ang. Parts-Of-Speech, w skrócie: POS); Rozpoznawanie nazwanych encji (NER); Częstość terminów - odwrotna częstość dokumentów (ang. Term Frequency-Inverse Document Frequency, w skrócie: TF-IDF). Źródło: Dang, N. C., Moreno-García, M. N., & De la Prieta, F. (2020). Sentiment Analysis Based on Deep Learning: A Comparative Study. Electronics, 9(3), 483.

Głębokie uczenie zrewolucjonizowało wiele obszarów, w tym rozpoznawanie mowy, analizę obrazów medycznych, samo-prowadzące się pojazdy, systemy rekomendacyjne, automatyczne tłumaczenie języków jak i wiele innych. Jego zdolność do modelowania złożonych danych i osiągania wybitnych wyników w zadaniach klasyfikacji, regresji i predykcji sprawia, że jest to jedno z najbardziej dynamicznie rozwijających się kierunków w dziedzinie sztucznej inteligencji. Powyższa grafika (rysunek 15) przedstawia dwa podejścia do klasyfikacji sentymentu: Machine Learning (ML) i Deep Learning (DL). Uczenie maszynowe to proces, który obejmuje ręczne przetwarzanie tekstu (tokenizacja, lematyzacja, itd.), ekstrakcję cech (TF-IDF, one-hot encoding) i modelowanie (Naive Bayes, SVM, regresja logistyczna). Natomiast uczenie głębokie daje możliwość automatyzacji ekstrakcji cech poprzez dense embeddings oraz używa sieci neuronowych z warstwami ukrytymi do modelowania. DL wykazuje bardziej zadowalające efekty, ponieważ automatycznie uczy się reprezentacji cech, wychwytuje wzorce o większej złożoności, jak również często modele te wykazują wyższą dokładność klasyfikacji. Warto jednak pamiętać, że rozwój uczenia głębokiego oraz transformacja z tym związana z wieloma wyzwaniami, np. zasobożernością, ponieważ do treningu głębokich sieci neuronowych potrzebne jest posiadanie specjalistycznego sprzętu oraz mocy obliczeniowej. Cechy takie jak automatyczność procesów może być ograniczeniem dla dziedzin takich jak prawo czy medycyna, gdzie oczekuje się weryfikowalności i odpowiedzialności, przez to w tych branżach modele te są krytykowane. Podsumowując, głębokie uczenie stanowi punkt zwrotny w rozwoju sztucznej inteligencji, oferując niezrównane możliwości modelowania danych. Pomimo wyzwań związanych z zasobami obliczeniowymi, przeuczeniem i interpretowalnością, kontynuowane badania i rozwój technologii mają na celu pokonanie tych barier, otwierając nowe horyzonty dla zastosowań AI.

### 2.2.1 Sztuczne Sieci Neuronowe

Sztuczne sieci neuronowe (ANN od ang. Artificial Neural Network) są podstawowymi elementami składowymi modeli głębokiego uczenia, często wykorzystywanymi do prostszych zadań lub jako komponenty w bardziej złożonych architekturach. W analizie nastrojów, ANN mogą służyć jako modele bazowe lub być częścią metod *ensemble*. Przykładem zastosowania w bieznesie będzie wykorzystanie podstawowych sieć ANN do analizy sentymentu tweetów przez firmę marketingową. Dane tekstowe na temat marek klientów przedsiębiorstwa są wstępnie przetwarzane i konwertowane do postaci wektorowej (np. TF-IDF), a następniem model jest trenowany na tym zestawie funkcji, ucząc się klasyfikować wpisy jako pozytywne lub negatywne. Chociaż wydajność może nie dorównywać bardziej wyrafinowanym modelom, prostota i szybkość ANN sprawiają, że są one realnym wyborem dla początkowych wysiłków związanych z analizą nastrojów, zapewniając szybki wgląd, który może kierować bardziej szczegółowymi badaniami.

Sieci neuronowe składają się z prostych, silnie połączonych jednostek przetwarzających, zwanych neuronami, z których każdy wykonuje dwie funkcje, mianowicie agregację swoich wejść od innych neuronów lub środowiska zewnętrznego oraz generowania wyjścia z agregowanych wejść (Dongare, Kharde, & Kachare, 2012). Podstawowym elementem ANN określa się uproszczony paradygmat biologicznego neuronu, inspiracja nadeszła z obserwacji funkcjonowania naturalnych sieci neuronowych w ludzkim mózgu. Sztuczne neurony, które symulują działanie swych anatomicznych odpowiedników wykonują agregację sygnałów wejściowych, które następnie są przetwarzane przy użyciu funkcji aktywacji w celu generowania sygnału wyjściowego. Wyjątkowa zdolność mózgu do nauki, adaptacji i rozpoznawania wzorców inspirowała naukowców do stworzenia matematycznych odpowiedników posiadających te same umiejętności. Sieci neuronowe wykazują zdumiewającą zdolność do uczenia się z danych, dostosowywania się do nowych, nieznanych wcześniej sytuacji i wykonywania złożonych obliczeń przy minimalnej ingerencji zewnętrznej. Umożliwia to ich zastosowanie w zadaniach, które dla tradycyjnych algorytmów obliczeniowych są trudne lub niemożliwe do realizacji, takich jak rozpoznawanie obrazów, przetwarzanie języka naturalnego, modelowanie sekwencji czasowych czy symulacje dynamicznych systemów. Warto podkreślić, że pomimo inspiracji żywymi organizmami jest wyraźna oraz jest częścią historii, ANN są odrębną kategorią narzędzi obliczeniowych, które wykorzystują matematyczną wiedzę statystyczną dla realizacji zadań informatycznych. Procesy adaptacyjne w ANN, realizowane przez algorytmy uczenia maszynowego, choć analogiczne do biologicznych procesów zachodzących w mózgu, opierają się na optymalizacji matematycznej i statystycznej analizie danych, a nie na dokładnym odtwarzaniu złożonych procesów biologiczno-chemicznych zachodzących w organizmie ludzkim.

### 2.2.2 Sztuczne Sieci Neuronowe Konwolucyjne

Sieci neuronowe konwolucyjne (CNN od ang. Convolution Neural Networks) są analogiczne do tradycyjnych ANN, ponieważ składają się z neuronów, które samo-optymalizują się poprzez uczenie. Sieci neuronowe konwolucyjne różnią się od innych form sztucznych sieci neuronowych tym, że zamiast skupiać się na całości dziedziny problemu, wykorzystywana jest wiedza o konkretnym typie wejścia. To z kolei pozwala na ustanowienie znacznie prostszej architektury sieci (O'Shea & Nash, 2015). Konwolucyjne sieci neuronowe różnią się od innych sieci neuronowych, tym, że charakteryzuje je unikatowa wydajność w przypadku, gdy dane wejściowe są obrazem, mową lub dźwiękiem. Specjalizacja CNN jest w obrębie rozpoznawania obrazów i zadań, które umożliwiają komputerom uzyskiwanie znaczących informacji z danych wizualnych (ang. computer vision). Znajdą one swoje zastosowanie między innymi w medycynie, gdzie będą wspomagać identyfikację tkanek nowotworowych przez lekarzy.

CNN są wysoce użyteczne w analizie setymentu, szczególnie z powodu ekstrakcji cech. Znane są one ze zdolności do wydajnej ekstrakcji lokalnych cech z danych wejściowych, które w przypadku tekstu mogą obejmować lokalne wzorce czy kombinacje słów. W kontekście NLP, pierwsze warstwy sieci konwolucyjnych mogą rozpoznawać podstawowe elementy tekstu, takie jak wyraz, podczas gdy głębsze warstwy mogą interpretować bardziej złożone struktury, takie jak frazy czy całe zdania, które są krytyczne dla rozumienia kontekstu i sentymentu. Jedną z najbardziej znaczących cech CNN, ma zdolność do automatycznego wydobywania istotnych cech, bez potrzeby ręcznego przygotowywania cech z tekstu przez specjalistów.

Aktualnie wiele firm realnie wykorzystuje Convolutional Neural Networks (CNN) w różnych aplikacjach. Amazon używa tych sieci w swojej usłudze Amazon Comprehend do analizy i klasyfikacji sentymentu. Oprogramowanie to jest częścią większej oferty Amazon Web Services (AWS) w zakresie przetwarzania języka naturalnego.

### 2.2.3 Rekurencyjne Sieci Neuronowe

Rekurencyjne sieci neuronowe (RNN od ang. Recurrent Neural Networks) to klasa sieci neuronowych, w których połączenia między neuronami tworzą skierowany cykl (rysunek 16), co tworzy pętle sprzężenia zwrotnego wewnątrz RNN (Dang, Moreno-García, & De la Prieta, 2020).

A diagram of a complex network

Description automatically generated

Rysunek 16. Rekurencyjna sieć neuronowa to sieć neuronowa o bardzo głębokim sprzężeniu zwrotnym, która ma warstwę dla każdego kroku czasowego. Jego wagi rozkładają się w czasie. Źródło: Sutskever, I. (2013). Training Recurrent Neural Networks. Doktorat, Graduate Department of Computer Science, University of Toronto.

Używają one danych sekwencyjnych lub szeregów czasowych, a swoje przeznaczenie odnajdują w tłumaczeniu języków (zastosowanie w *Google Translate*), przetwarzania języka naturalnego, ale także do zadań takich jak rozpoznawanie mowy (zastosowanie w wirtualnym asystencie dla użytkowników firmy *Apple – Siri*).

Jako najbardziej znane typy sieci neuronowych uznaje się prostą rekurencyjna sieć neuronowa (RNN), Long Short-Term Memory Neural Network (LSTM) i Gated Recurrent Unit (GRU). To te kolejne rozwiązania stanowią kroki w rozwoju dziedziny sztucznej inteligencji, poszerzając naszą wiedzę na temat możliwości sztucznych sieci neuronowych, a przez to odnajdując następne praktyczne rozwiązania. Warto tutaj podkreślić, jak dalsze badania w tej dziedzinie mogą dać szansę na przełomowe odkrycia, udogodnienia i pomoc w lepszym rozumieniu otaczającego nas świata.

Architektura Recurrent Neural Networks (RNN) została wprowadzona przez Davida Rumelharta w 1986 roku. RNN zawierają wewnętrzne pętle, umożliwiające im przetwarzanie danych sekwencyjnych w czasie i wykorzystywanie informacji z przeszłości do wpływania na bieżące i przyszłe przetwarzanie danych. Sieci te posiadają wielowymiarowe stany ukryte o nieliniowej dynamice, pełniąc rolę pamięci sieci (Sutskever, Martens i Hinton, 2011; Mikolov, Joulin, Chopra, Mathieu i Ranzato, 2014). RNN mogą mapować sekwencje wejściowe na sekwencje wyjściowe w bieżącym kroku czasowym i przewidywać przyszłe sekwencje (Salehinejad, Sankar, Barfett, Colak i Valaee, 2018). W przeciwieństwie do sieci ze sprzężeniem zwrotnym (ang. feedforward networks), RNN wykorzystują pętle sprzężenia zwrotnego do kontekstowego przetwarzania sekwencji, pomagając w zrozumieniu złożonych sygnałów w długich okresach, takich jak gramatyka i znaczenie w analizie języka naturalnego.

Trening RNN obejmuje propagację w przód (ang. Forward Propagation) i propagację wsteczną w czasie (ang. Backpropagation Through Time, w skrócie BPTT), która rozszerza standardową propagację wsteczną w krokach czasowych. Ten proces uczenia jest złożony ze względu na potencjalne problemy, takie jak zanikanie lub eksplodowanie gradientów oraz złożoność obliczeniowa (Lipton, Berkowitz i Elkan, 2015; Sherstinsky, 2020).

Znaczący postęp nastąpił pod koniec lat 90. XX wieku wraz z opracowaniem sieci pamięci długoterminowej (ang. Long Short-Term Memory network, w skrócie LSTM) przez S. Hochreitera i J. Schmidhubera. LSTM rozwiązują ograniczenia RNN poprzez poprawę zdolności uczenia się długoterminowych zależności, poprawę wydajności w przetwarzaniu języka naturalnego i rozpoznawaniu mowy (Hochreiter i Schmidhuber, 1997).

A diagram of a cell block

Description automatically generated

Rysunek 17. Przykład sieci z ośmioma jednostkami wejściowymi, czterema jednostkami wyjściowymi i dwoma blokami komórek pamięci o rozmiarze 2. *in1* oznacza bramkę wejściową, *out1* oznacza bramkę wyjściową, a *cell1/block1* oznacza pierwszą komórkę pamięci bloku 1. Źródło: Hochreiter, S., & Schmidhuber, J. (1

997). Long Short-Term Memory. Neural Computation, 9(8), 1745.

Podstawowym elementem składowym sieci Long Short-Term Memory jest komórka, zaproponowana przez Hochreiter’a i Schmidhuber’a. Na rysunku 17 widnieje przykład sieci. Poprzez *in1* oznaczono bramkę wejściową, *out1* oznacza bramkę wyjściową, a *cell1/block1* oznacza pierwszą komórkę pamięci bloku 1. Architekturą komórki pamięci (ang. memory cel) jest cel/block1, która zakłada gęste połączenia między każdą jednostką bramki/ każdą komórką pamięci, a wszystkimi jednostkami niebędącymi jednostkami wyjściowymi. Przepływ błędu następuje poprzez połączenia do jednostki wyjściowej oraz poprzez stałe połączenia wewnętrzne w blokach komórek (ang. cel blocks), proces ten zastępuje z wykorzystaniem efektywnej, przeciętnej reguły aktualizacji. Przepływ błędu jest przycinany, gdy tylko „chce” opuścić komórki pamięci lub jednostki bramek. Dlatego żadne pokazane powyżej połączenie nie służy do propagowania błędu z powrotem do jednostki, z której połączenie to pochodzi (z wyjątkiem połączeń do jednostek wyjściowych), chociaż same połączenia są modyfikowalne (Hochreiter & Schmidhuber, 1997). Koncepcja ta nie propaguje błędu z powrotem do tej jednostki, z której ono wyszło, pomimo, że połączenie jest w rzeczy modyfikowane. W ten sposób nastąpiła poprawa umiejętność zapamiętywania w standardowej komórce rekurencyjnej poprzez wprowadzenie „bramki” (ang. gate) do komórki. Jednak przez lata dokonano wiele modyfikacji tego pionierskiego rozwiązania. Odmiany obejmują LSTM bez bramki zapominającej, LSTM z bramką zapominającą i LSTM z połączeniem przez wizjer. Zwykle termin komórka LSTM oznacza LSTM z bramką zapominającą. Najpierw przedstawiamy oryginalny model LSTM, który posiada tylko bramki wejściowe i wyjściowe (Yu, Si, Hu & Zhang, 2019).

A diagram of a computer system

Description automatically generated

Rysunek 18. Oryginalna architektura LSTM. Źródło: Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. Neural Computation, 1–36.

Komórka pamięci długoterminowej (rysunek 18) jest przeznaczona do przetwarzania danych sekwencyjnych, takich jak szeregi czasowe, nagrania audio lub tekst. Składa się z kilku komponentów, które wspólnie zarządzają przechowywaniem, aktualizacją i przesyłaniem informacji. Oryginalna architektura LSTM obejmuje bramkę wejściową i bramkę wyjściową (ang. input and output gate).

Bramka wejściowa określa, jakie nowe informacje mają być przechowywane w stanie komórki, przy użyciu funkcji aktywacji sigmoidalnej i tanh. Funkcja sigmoidalna decyduje, jakie informacje zachować, a które odrzucić, natomiast funkcja tanh dodaje lub odejmuje nowe informacje do i ze stanu komórki.

Stan komórki (ang. cell state) działa jako główny nośnik informacji, przechowując informacje zarówno krótkoterminowe, jak i długoterminowe. Stan komórki aktualizuje się poprzez połączenie starego stanu, przeskalowanego przez bramkę zapominania, z nowymi wartościami kandydującymi przefiltrowanymi przez bramkę wejściową.

Brama wyjściowa decyduje, jakie informacje wysłać na podstawie stanu komórki, ułatwiając sieci zachowanie ważnych informacji i reagowanie na bieżące wejścia. Wykorzystuje funkcję sigmoidalną do filtrowania informacji o stanie komórki, które są następnie normalizowane przez funkcję aktywacji tanh (rysunek 19) przed przekazaniem na wyjście.

Architektura ta umożliwia LSTM selektywne przesyłanie ważnych informacji, zapewniając wydajne przetwarzanie i przechowywanie danych sekwencyjnych.

A diagram of a computer system

Description automatically generated

Rysunek 19. Architektura LSTM z bramką zapominania. Źródło: Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. Neural Computation, 1–36.

Gers, Schmidhuber i Cummins (2000) zmodyfikowali oryginalny LSTM w 2000 roku, wprowadzając do komórki bramkę zapominania (Yu, Si, Hu & Zhang, 2019). Bramka zapominania (ag. forget gate), decyduje o tym jakie informacje z poprzedniego stanu komórki będą zapomniane. Jest to dokonywane poprzez zastosowanie sigmoidalnej funkcji aktywacji. LSTM rozwiązuje problem zanikającego gradientu, dzięki zdolności zapamiętywania i zapominania informacji. Zastosowane tu mechanizmy bramek zapewniają adaptacyjność na dynamicznie zmieniające się dane ze środowiska, co sprawia, że są elastyczne(Hochreiter & Schmidhuber, 1997).

Pomimo zalet sieci neuronowe LSTM są powolne ze względu na dużą równoległość i sekwencyjny charakter, ale może zostać to rozwiązane poprzez zaprezentowanie w najbliższej przyszłości niezawodnego akceleratora sprzętowego (Smagulova & James, 2019).

W 2014 roku Cho i współpracownicy wprowadzili bramkowane jednostki rekurencyjne (ang. Gated Recurrent Unit, w skrócie: GRU) jako innowację w rekurencyjnych sieciach neuronowych. GRU upraszczają architekturę pamięci długoterminowej (Cho, van Merriënboer, Bahdanau i Bengio, 2014), zachowując podobną lub lepszą wydajność w niektórych scenariuszach i zmniejszając złożoność obliczeniową. Mają na celu przechwytywanie zależności w różnych skalach czasowych z mniejszą liczbą bramek (rysunek 20): bramki aktualizacji i bramki resetowania (Chung, Gulcehre, Cho i Bengio, 2014; Tüske i in., 2016).

A diagram of a computer system

Description automatically generated

Rysunek 20. Architektura LSTM z GRU. Źródło: Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. Neural Computation, 1–36.

Bramka aktualizacyjna (ang. update gate) określa, ile informacji z przeszłości należy przenieść do przyszłych kroków. Działa podobnie do połączonych bramek zapominania i wprowadzania w LSTM, decydując, które informacje zachować w stanie komórki i jakie nowe informacje dodać.

Natomiast bramka resetująca (ang. reset gate) kontroluje, ile informacji z przeszłości należy zapomnieć, co jest niezbędne do modelowania zależności między odległymi punktami w sekwencjach. Bramka ta pomaga modelowi w selektywnym resetowaniu części stanu komórki podczas przetwarzania nowych danych wejściowych, przechwytując w ten sposób zależności krótkoterminowe.

Przepływ informacji rozpoczyna się, gdy dane wejściowe i stan poprzedni przechodzą przez obie bramki, dynamicznie regulując przepływ informacji. GRU tworzy nowy stan kandydujący w oparciu o przefiltrowane informacje z przeszłości (biorąc pod uwagę bramkę resetującą) i nowe dane wejściowe. Ten stan kandydujący jest następnie wykorzystywany do obliczenia bieżącej zawartości pamięci, która łączy przekształcony poprzedni stan ukryty i bieżący stan wejściowy, znormalizowany za pomocą funkcji aktywacji tanh (Mortezapour Shiri, Perumal, Mustapha i Mohamed, 2023).

GRU aktualizują i resetują stan ukryty przy mniejszej liczbie bramek niż LSTM, dzięki czemu są szybsze i wydajniejsze. Ta prostota pozwala GRU szybciej przetwarzać większe zbiory danych, zużywać mniej energii i być odpowiednia dla urządzeń o ograniczonych zasobach obliczeniowych (Zegarra i Vargas-Machuca, 2023). Podczas gdy LSTM mogą mieć przewagę w zadaniach wymagających długotrwałych zależności ze względu na ich bardziej złożoną strukturę, GRU oferują równowagę wydajności i wydajności obliczeniowej, dzięki czemu są skuteczne w wielu zadaniach przetwarzania języka naturalnego (Ni i Cao, 2020).

GRU są szeroko stosowane w przetwarzaniu języka naturalnego ze względu na ich wysoką wydajność i łatwość modyfikacji, szczególnie podczas pracy z mniejszymi zbiorami danych. Ich prostsza architektura w porównaniu z LSTM pozwala na szybsze szkolenie i wdrażanie, dzięki czemu nadają się do zastosowań i scenariuszy czasu rzeczywistego o ograniczonej mocy obliczeniowej.

## 2.3 Algorytmy Przetwarzania Języka Naturalnego

Przetwarzanie języka naturalnego (ang. Natural Language Processing, w skrócie NLP) stoi w połowie drogi pomiędzy informatyką i lingwistyką obliczeniową i zajmuje się konwersją pisanego i mówionego naturalnego języka ludzkiego na ustrukturyzowane dane, które można wydobywać (Fanni, Febi, Aghakhanyan, & Neri, 2023). Ma na celu umożliwić komputerom interpretacje, zrozumienie jak również może generować tekst podobny do tego jakim posługują się na co dzień ludzie. Zadanie to ma być wykonane w sposób użyteczny, jak również jak najbardziej przybliżony do realnego sposobu komunikacji. W ostatnich latach wraz z rozwojem Internetu zadanie zyskało na znaczeniu, ponieważ zauważona została potrzeba szybkiej reakcji na komunikaty, automatyzacji zadań, rozpoznawania mowy czy tłumaczenia wiadomości tekstowych. Rozwój NLP jest napędzany przez rosnące potrzeby technologiczne, globalizację oraz wymagania biznesu. Narzędzia i podejścia NLP można stosować na różnych poziomach analizy. Poziomy analizy językowej rozciągają się od najniższego, fonologicznego, po najwyższy, pragmatyczny (Crowston, Allen, & Heckman, 2011). Najniższy poziom koncentruje się na dźwiękach języka i może obejmować rozpoznawanie mowy, analizę intonacji czy nagrań. Najwyższy poziom, który bada pragmatyczność to obszar badań wpływu kontekstu na intencję, ton, znaczenie wypowiedzi. A pomiędzy tymi poziomami, NLP skupia się na strukturze słów, ich znaczeniu, czy na organizowaniu zdań w komunikatach.

### 2.3.1 Przetwarzanie wstępne i reprezentacja tekstu

Kluczowym wyzwaniem stojącym przed NLP jest złożoność ludzkiego języka, jego ambiwalencja, sarkazm, bogactwo kontekstu, czy brak regularności. Dodatkowo, warto zauważyć, że codziennością w komunikatach tekstowych stają się obrazki, tzw. *Emoji*, które stanowią dodatkową informację o emocjach przekazywanych przez autora. Wielopoziomowość zagadnienia sprawia, że zrozumienie tekstu nie jest łatwa i jednoznaczna dla wszystkich osób, z wszystkich grup wiekowych czy kręgów kulturowych. Dlatego komputery wykonujące analizę słów i struktury muszą wykazać się głębokim zrozumieniem semantyki, pragmatyki oraz kontekstu nadawcy.

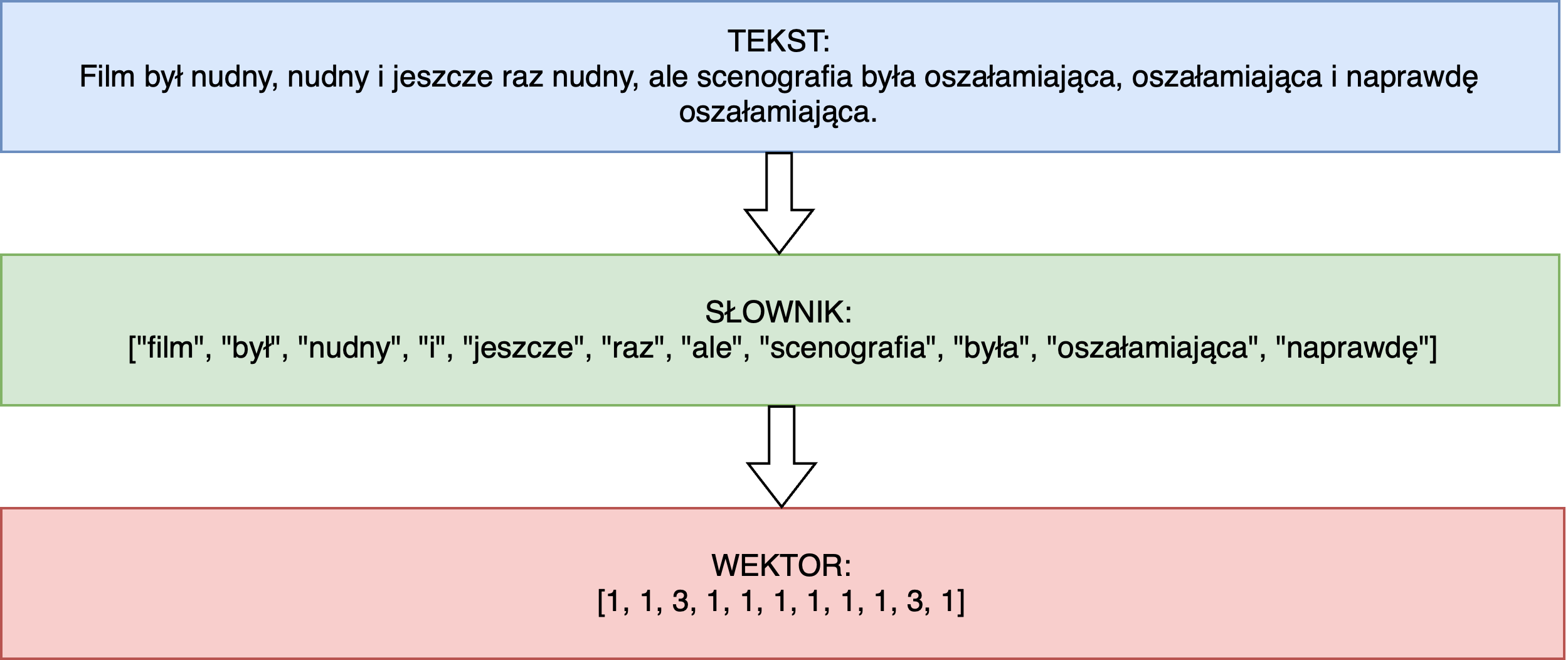
Proces analizy tekstu można podzielić na kilka głównych etapów, które rozpoczyna wstępne przetwarzanie, a następnie w zależności od potrzeb może nastąpić: normalizacja, wektoryzacja, redukcja wymiarowości, ekstrakcja cech, modelowanie i ewaluacja modelu. Te kolejne kroki przedstawiają wymaganą kompleksowość podejścia do problematyki. Każdy etap jest niezbędny do efektywnego analizowania zadanych danych.

Tokenizacja to podstawowy proces w przetwarzaniu języka naturalnego, który wykonuje podział podanych danych tekstowych na mniejsze jednostki, zwane tokenami oraz eliminacji białych znaków, interpunkcji, czy innych symboli uznanych za nieistotne z punktu widzenia podanego problemu. Tokeny mogą być słowami, frazami lub pojedynczymi znakami. Celem tego procesu jest dyskretyzacja elementów, aby umożliwić łatwiejszą analizę przez kolejne algorytmy.

Usuwanie słów nieistotnych (ang. stop-words), to kolejne zadanie, które polega na pozbyciu się powszechnych słów niewnoszących informacji z punktu widzenia analizy sentymentu, mogą być to wyrazy takie jak: „ale”, „oraz”, „i”. Wykonanie tego kroku jest niebywale istotne, ponieważ pozwala zredukować rozmiar danych, co wspiera efektywność przetwarzania. Następnym etapem może być stemming, czyli proces redukcji słów do ich podstawowej formy. Stemming jest stosowany głównie w wyszukiwaniu informacji czy klasyfikacji, ponieważ umożliwia grupowanie. Natomiast lematyzacja, prowadzi do przypisania słowom ich bazowych form (lematów), jednocześnie biorąc pod rozwagę ich znaczenie w tym kontekście. Proces ten jest bardziej złożony od stemmingu, gdyż wymaga pełnej analizy morfologicznej, struktury gramatycznej języka. Znajduje ona swoje szczególne zastosowanie w zadaniach, takich jak tłumaczenie maszynowe, czy aplikacje do zrozumienia języka naturalnego.

Kolejnym krokiem we wstępnym przetwarzaniu danych języka naturalnego jest transformacja tekstu na format numeryczny, czyli wektoryzacja. Proces ten umożliwia stworzenie danych, które mogą być konsumowane przez model uczenia maszynowego i wykonanie operacji na nich. Wyróżnia się kilka metod wektoryzacji: Bag of Words (BoW), Term Frequency (TF), Term Frequency-Inverse Document Frequency (TF-IDF), embeddingi.

Bag of Words to metoda polegająca na reprezentowaniu tekstu jako zestawu słów, ale nie uwzględniając ich kolejności w zadanym tekście, każde pojedyncze słowo jest traktowane jako osobna cecha w wektorze stworzonym ze słownika. A następnie każda pozycja utożsamiana jest z liczbą wystąpień pojedynczego słowa w danym dokumencie za pomocą liczby (rysunek 21). W modelu worka słów dokumenty są przedstawiane jako nieuporządkowany zbiór słów, bez względu na gramatykę i kolejność słów, co czyni je podatnymi na dwuznaczność (Jurafsky & Martin, 2019). Metoda ta charakteryzuje się prostotą, ale co za tym idzie – ma ograniczenia związane z utratą informacji o kontekście czy strukturze języka.



Rysunek 21. Przykładem zastosowania metody Bag of Words. Źródło: opracowanie własne

Kolejną techniką wspomagającą wektoryzację tekstu jest miara Term Frequency (TF), która wyraża jak często określone słowo pojawia się w danym dokumencie. Częstotliwość występowania terminu w dokumencie to po prostu liczba wystąpień terminu podzielona przez całkowitą liczbę terminów w dokumencie (Manning, Raghavan & Schütze, 2009).

(2.2)

gdzie:

= miara term frequency, czyli częstotliwość terminu *t* w dokumencie *d*

*t* = określony termin/ słowo

*d* = określony dokument

= liczba wystąpień słowa *t* w dokumencie *d*

*sumd* = to suma wszystkich słów w dokumencie

Warto zauważyć, że TF zaczyna budować intuicję, że częste terminy są ważne, ale samo liczenie słów nadaje zbyt dużą wagę nieistotnym słowom, takim jak „*the*” czy „*jest*” (Jurafsky & Martin, 2019). Term Frequency jest szeroko stosowany w wielu aplikacjach klasyfikacji tekstu, gdzie może służyć w identyfikacji treści często występujących w dokumentach, czy też jako część algorytmu TF-IDF.

Ideą Term Frequency-Inverse Document Frequency (w skrócie: TF-IDF) jest zmniejszenie wpływu słów, które często występują w wielu dokumentach, a jednocześnie nie wpływają na większe zrozumienie treści przekazywanej. Ważenie TF-IDF neguje wpływ wysokiej częstotliwości terminów w przypadku terminów pojawiających się w wielu dokumentach, równoważąc w ten sposób znaczenie częstotliwości terminów z rzadkością w całym korpusie (Manning, Raghavan & Schütze, 2009). Ta statystyczna miara jest obliczana jako iloczyn TF i IDF. *Inverse Document Frequency* (w skrócie IDF) to logarytm stosunku liczby wszystkich dokumentów *N*, do liczby dokumentów *df(t)* zawierających słowo *t*. Wzór dla IDF jest zatem następujący:

(2.3)

*IDF(t) = log* ()

gdzie:

*IDF(t)* = miara IDF, która mierzy jak ważne jest słowo w korpusie dokumentów

*N* = całkowita liczba dokumentów w analizowanym korpusie dokumentów

*t* = określony termin/ słowo

Natomiast całość może być wyrażona poprzez:

(2.4)

*TF-IDF (t, d) = TF (t, d) IDF(t)*

gdzie:

*TF-IDF (t, d)* = miara TF-IDF, czyli wynik TF-IDF dla terminu *t* w dokumencie *d*

*t* = określony termin/ słowo

*d* = określony dokument

*TF (t, d)* = miara term frequency, czyli częstotliwość terminu *t* w dokumencie *d*

*IDF(t)* = miara IDF, która mierzy jak ważne jest słowo w korpusie dokumentów

Miara ta jest często wykorzystywana w systemach wyszukiwania czy odzyskiwania informacji, gdzie umożliwia klasyfikację, modelowanie, wraz z osłabieniem znaczenia słów powszechnie występujących, ale wielokroć nie noszącym za sobą wpływu na przekaz treści.

Następną techniką przetwarzania języka naturalnego jest wektoryzacja semantyczna, inaczej zwana embedding tekstowy. W przeciwieństwie do TF czy BoW, czyli metodach opartych na zliczaniu embeddingi przechowują więcej informacji o kontekście. Najbardziej popularne metody to Word2Vec i GloVe. Word2Vec zostało stworzone przez firmę Google i wykorzystuje sieci neuronowe w tworzeniu wektorów słów. Word2Vec to wydajny obliczeniowo model predykcyjny do uczenia się osadzania słów na podstawie surowego tekstu. Występuje w dwóch wersjach: Continuous Bag of Words (CBOW) i modelu Skip-Gram, przy czym oba uczą się przewidywać słowa na podstawie kontekstu na różne sposoby (Mikolov, et al.,2013). GloVe to algorytm uczenia się bez nadzoru, służący do uzyskiwania reprezentacji wektorowych słów poprzez agregowanie globalnych statystyk współwystępowania słów z korpusu. Powstałe reprezentacje przedstawiają interesujące liniowe podstruktury przestrzeni wektorowej słowa (Pennington, Socher & Manning, 2014). W obu metodach modele są trenowane na korpusach tekstowych, gdzie uczenie wykonywane jest na reprezentacji wektorów słów na podstawie ich współwystępowania z innymi słowami (Ghorpade & Mane, 2017). Wygenerowane wektory mogą mieć od kilkudziesięciu do kilkuset wymiarów, w zależności od złożoności modelu, jak również rozmiaru danych, na których nastąpiło trenowanie. Słowa o podobnej semantyce po procesie szkolenia modelu będą mieć bliskie sobie wektory, co daje unikatową możliwość wykrywania synonimów, a dalej do zautomatyzowanego tłumaczenia. Podczas gdy tradycyjne metody mogą prowadzić do problemu wielowymiarowości, embeddingi pozwalają na stałą liczbą wymiarów podczas reprezentacji słów w dokumencie. Wyjątkową cechą tego podejścia jest przenośność, co znaczy, że wektory słów wytrenowane na jednym zestawie danych mogą być wykorzystane do innych zadań, co nadaje im elastyczności. Embedding dokumentów są przydatne przy zadaniach takich jak klasyfikacja dokumentów, analiza sentymentu i grupowanie dokumentów, ponieważ skutecznie oddaje znaczenia semantyczne i struktury tematyczne (Le & Mikolov, 2014).

### 2.3.2 Redukcja wymiarowości i jej wpływ na Analizę Sentymentu

Redukcja wymiarowości jest kluczowym aspektem w procesie analizy informacji, dla danych o wysokiej złożoności. Proces ten polega na zminimalizowaniu liczby losowych zmiennych branych pod uwagę, a stworzenie zestawu głównych zmiennych. Celem tego jest uproszczenie zestawu danych bez utraty informacji, które przekazują. Wielowymiarowość jest powodem zwiększenia czasu obliczeń, czy większych zapotrzebowań pamięciowych. Może prowadzić także do nadmiernego dopasowania modelu przez powstanie szumu. Jest również utrudnieniem, gdy użytkownik chce wykonać wizualizacje, poprzez występujące wzorce, które są skomplikowane w interpretacji.

Pierwszą techniką, która zostanie przedstawiona w tej pracy będzie analiza składowych głównych (ang. Principal Component Analysis, w skrócie PCA). PCA zmniejsza wymiarowość, znajdując nowy zestaw zmiennych, mniej liczny, ale zawierający większość informacji z pierwotnego zestawu (Jolliffe, 2002). W ramach analizy danych pochodzących z mediów społecznościowych, zastosowanie analizy składowych głównych (PCA) umożliwia wyróżnienie istotnych cech oraz izolację kluczowych czynników wpływających na sentyment wpisów użytkowników, jednocześnie redukując informacje do tych o największym znaczeniu oraz pomagając w identyfikacji najbardziej dyskryminujących fraz. Zminimalizowanie problemów związanych z tzw. „przekleństwem wymiarowości” przyczynia się do zwiększenia wydajności działania klasyfikatorów sentymentu. W opisywanym kontekście, analiza składowych głównych może pomóc zidentyfikować, które zmienne mogą być silnie skorelowane z typem sentymentu. Uświadomienie sobie tych zależności może być kluczowe dla obrania właściwego podejścia podczas tworzenia predyktora.

Kolejną techniką redukcji wymiarowości jest rozkład według wartości osobliwych (ang. Singular Value Decomposition, w skrócie SVD), która jest metodą faktoryzacji macierzy. SVD zapewnia najlepsze przybliżenia macierzy niższego rzędu, które można wykorzystać do takich zadań, jak kompresja obrazu i redukcja szumów (Deerwester, et al.,1990). SVD w odróżnieniu do PCA stanowi istotny element *latent semantic analysis* (LSA), który wykorzystuje tę technikę do wydobywania ukrytych wzorców i relacji między terminami, a dokumentami w danym zbiorze danych. SVD wykonuje swoje zadanie w ujawnianiu ukrytych struktur semantycznych w danych pochodzących z mediów społecznościowych. Predysponuje to badaczy do znacznego pogłębienia swojej wiedzy oraz dokładnego zrozumienia tematu koncepcji i nastrojów autorów danych wpisów. Przez redukcje wymiarowości, wraz z jednoczesnym zatrzymaniem najistotniejszych semantycznych cech, ujawniony zostaje ogólny sentyment, jak również badacz dostaje możliwość wgłębienia się w sposoby subiektywnej oceny w różne aspekty podejmowanych tematów.

W kontekście NLP, redukcja złożoności jest stosowana do analizy tekstów, może pomóc identyfikacji podobieństw między różnymi typami dokumentów (np. wiadomościami), co ułatwi generowanie trafniejszych rekomendacji. Uproszczone dane są również łatwiejsze do klastrowania i klasyfikacji, gdzie teksty są kategoryzowane na podstawie ich tematycznej zawartości lub sentymentu. Zarówno PCA, jak i SVD mogą przyczynić się do lepszego i szybszego trenowania modeli klasyfikacyjnych, ponieważ modele te będą pracować na mniejszej liczbie cech, które są bardziej znaczące, a także umożliwi tworzenie bardziej przejrzystych wizualizacji w kolejnych etapach pracy.

### 2.3.3 Metody Analizy Sentymentu

Analiza sentymentu wykorzystuje różne technologie do oceny tekstu. Można wyróżnić podejście deterministyczne i probabilistyczne. Deterministyczne podejście opiera się na z góry określonych regułach, które klasyfikują dokumenty na podstawie określonych słów czy fraz, w ten sposób przypisując wartość sentymentu. Techniki te są często oparte na zestawach słów kluczowych lub zdefiniowanych strukturach gramatycznych. Ich główną zaletą jest przewidywalność i spójność wyników, jednak są one mniej elastyczne, przez co mogą nie radzić sobie dobrze w przypadku subtelnie wyrażonych emocji, które nie mieszczą się w ramach zdefiniowanych reguł. Podstawowe podejście tego typu to analiza sentymentu oparta na słownikach, reguły gramatyczne, wykrywanie specyficznych wyrażeń emocjonalnych. Przykładem może być system, który analizuje recenzje produktów lub usług, przypisując punkty za każde pozytywne czy negatywne słowo, aby uzyskać końcową ocenę sentymentu tekstu.

Natomiast probabilistyczne podejście opiera się na uczeniu maszynowym, używając danych treningowych do wykrywania sentymentu. Techniki te mogą obejmować różne rodzaje klasyfikatorów, takich jak modele Bayesa, drzewa decyzyjne, czy sieci neuronowe, które probabilistycznie oceniają prawdopodobieństwo przynależności tekstu do danej kategorii sentymentalnej. Zaletą tych metod jest zdolność do modelowania złożonych i subtelnych wyrażeń emocjonalnych, które mogą umykać deterministycznym systemom. Wadami są natomiast wyższe wymagania obliczeniowe i potrzeba dużych, dobrze oznakowanych zbiorów danych treningowych. Przykładem takie podejścia jest wykorzystanie analiza sentymentu tweet’ów w czasie rzeczywistym, gdzie modele są trenowane, aby rozpoznawać na zmieniające się nastroje społeczne. Metody probabilistyczne, które używają teorii prawdopodobieństwa do przewidywania zjawisk, posiadają zdolność adaptacji do nowych zestawów danych bez potrzeby manualnej modyfikacja istniejących reguł. Jest to niezmiernie istotne w dynamicznie rozwijającym się środowisku, a takim właśnie są portale internetowe. Systemy przetwarzania języka naturalnego dla danych pochodzących ze stron internetowych są tego bardzo dobrym przykładem, ponieważ muszą się one nieustannie uaktualniać, aby z precyzją interpretować nowe komentarza, które będą mieć bieżące konteksty i prawdopodobnie będą posiadać drobne zmiany w słownictwie. Języki są żywymi istotami, stale się rozwijającymi i zmieniającymi się, aby dopasować się do potrzeb komunikacyjnych ich użytkowników (Pinker, 2007). W miarę zmian języka w miarę upływu czasu, w tym pojawiania się nowego slangu, żargonu technicznego i zmian w użyciu słów, modele uczenia maszynowego do przetwarzania języka muszą stale się dostosowywać, aby zachować ich przydatność i skuteczność (Eisenstein, 2019). Systemy monitorujące napisane opinie w Internecie są znakomitym przykładem, ponieważ są one zmuszone do ciągłego doszkalania nowych postów i komentarzy, aby lepiej zrozumieć tekst, a także niuanse w wyrażaniu nastroju. Probabilistyczne podejścia są przystosowane do modelowania niepewności i niejednoznaczności w danych, co nadaje im elastyczności. Umiejętność aktualizacji i skalowalność w miarę pojawiania się nowych słów lub zmieniających się wzorców użycia jest nieocenione w przypadkach, gdy występują niuanse językowe i następuje zmiana kontekstu.

Zarówno deterministyczne, jak i probabilistyczne podejścia mają swoje miejsce w analizie sentymentu, a wybór odpowiedniej metody często zależy od specyfiki zadania, dostępności danych oraz wymagań co do dokładności i złożoności analizy.

### 2.3.4 Graficzne przedstawienie danych w Przetwarzaniu Języka Naturalnego

Wizualizacja danych to proces ich prezentacji graficznej, która umożliwia odbiorcom głębsze zrozumienie problematyki, analizowani i interpretacje informacji, które mogą charakteryzować się wysoką złożonością. W zagadnieniach z obszaru przetwarzania języka naturalnego, dane są często skomplikowane, obszerne, przez co wykonanie ilustracji, jest metodą wytłumaczenia ich i umożliwia wydobycie ukrytych wzorców, trendów, relacji.

Istnieje wiele narzędzi, które mają na celu wspomaganie analizy eksploracyjnej danych. Pierwszą z nich jest chmura słów (ang. word cloud), która prezentuje słowa ze zbioru danych, tak, aby wielkość słowa była proporcjonalna do częstotliwości występowania. Jest to jedna z najpopularniejszych metod do szybkiego identyfikowania przeważających terminów. Techniki wizualizacji, takie jak chmury słów, mogą szybko ujawnić najważniejsze wyrazy w zbiorze danych, oferując natychmiastowy obraz dominujących tematów w tekście (Manning, Raghavan & Schütze, 2009). Kolejną metodą są mapy ciepła (ang. heatmap), które przedstawiają współwystępowanie między różnymi terminami w korpusie. Są one szczególnie przydatne przy analizie macierzy korelacji. Gdy autor chce pokazać strukturę gramatyczną dokumentu, czy relacje zależności między wyrazami, można zastosować wtedy wykresy zależności. Natomiast w przypadku analizy embeddingu tekstowego, warto zapoznać się z projekcją wektorów słów (ang. word vector projection), ponieważ mogą one pokazać, jak wyrazy są grupowane, oceniać prawdopodobieństwa między słowami.

Wizualizacja danych to dziedzina o znaczącej mocy dla biznesu. Wyobraźmy sobie sytuację, w której firma telekomunikacyjna otrzymywała wiele skarg na temat swoich usług. W celu zidentyfikowania głównych wad wykonano przetwarzanie języka naturalnego danych, a następnie stworzono wizualizacje. Chmura słów mogłaby ujawnić główne problemy jak „oczekiwanie” i „ceny”. Wykres sentymentu pokazałby procentowy rozkład opinii: 40% pozytywnych, 30% neutralnych, 30% negatywnych. Natomiast heatmapa emocji wykazałaby intensywność emocji takich jak frustracja, zadowolenie czy złość. Na podstawie tych analiz można podjąć działania prowadzące do konkretnych i wymiernych korzyści biznesowych.

## 2.4 Regularyzacja

Próba zbyt dobrego pobrania parametrów i usunięcia szumu z danych treningowych skutkuje słabą wydajnością na nowych danych. Tutaj właśnie pojawiają się techniki regularyzacji. Regularyzacja, to proces dodania kary do funkcji kosztu, używanej podczas treningu modelu. Poprzez kontrolowanie wartości parametrów modelu, kontroluj złożoność, co sprzyja tworzeniu modeli o prostszej budowie. Jest to niezwykle ważne, ponieważ prowadzi ten proces do zminimalizowania prawdopodobieństwa overfittingu, a model będzie lepiej generalizował. Dodanie regularyzacji jest strategiczną decyzją i dlatego została dołączona do potoku budowania modelu.

Podstawową ideą wszystkich metod regularyzacji jest ograniczenie przestrzeni możliwych rozwiązań (Poggio, Torre, & Koch, 1985). Może obejmować techniki takie jak L1, L2 lub ich kombinacje (np. Elastic Net). Regularyzacja jest jednym z kluczowych elementów uczenia maszynowego, zwłaszcza głębokiego uczenia się (Goodfellow et al., 2016). Jest to dowolna technika uzupełniająca, której celem jest lepsze uogólnianie modelu, tj. generowanie lepszych wyników na zbiorze testowym (Kukačka, Golkov, & Cremers, 2017). Gdy sieci neuronowe uczą się reprezentować złożone relacje między danymi wejściowymi i tym co jest podawane na wyjściu może dojść do nadmiernego dopasowania modelu. O nadmiernym dopasowaniu mówimy, gdy sieć neuronowa, uczy się wysoce złożonej reprezentacji, modelując z wysoką dokładnością, lecz z małą uniwersalnością. Nadmierne dopasowanie to użycie modeli lub procedur, które naruszają oszczędność, to znaczy obejmują więcej terminów, niż jest to konieczne, lub zastosuj bardziej skomplikowane podejścia, niż jest to konieczne (Hawkins, 2004). W rezultacie działa wyjątkowo dobrze na zbiorze danych treningowych, ale słabo uogólnia zależności na danych testowych, które nie były wcześniej znane. Słaba wydajność dla nowych danych pojawia się także, jeśli model ma dużą złożoność, błąd uczenia może się zmniejszyć, ale błąd testowy nie ulegnie poprawie. Dla rozwiązania właśnie tego problemu pojawiły się zastosowania z technikami regularyzacji. Dokonują one niewielkich modyfikacji algorytmu uczenia się, aby stworzyć model o lepszych umiejętnościach uogólniania, a co za tym idzie lepszej predykcji. W uczeniu maszynowym regularyzacja karze parametry modelu, które są dostosowywane w procesie uczenia, aby najlepiej pasowały do danych treningowych. W głębokim uczeniu penalizuje ona faktycznie macierze wag węzłów (Jain, 2018) przypisanych do połączeń między neuronami. Regularyzacja dodaje karę do funkcji kosztu modelu za zbyt duże wartości współczynników lub wag. W wyniku takiej penalizacji, zachęca ona model do większej generalizacji. Istnieją różne metody regularyzacji: regularyzacja Lasso (L1), Ridge (L2), wczesne zatrzymywanie, powiększanie danych (ang. data augmentation), dodawanie szumu, dropout, wczesne zatrzymanie (ang. early stopping). W pracy tej zostaną opisane tylko wybrane metody.

Regularyzacja L1 i L2 to dwie popularne techniki, które działają poprzez dodanie dodatkowego terminu do funkcji kosztu, który penalizuje duże wartości współczynników.

Regularyzacja L1, znana również pod nazwą Lasso (ang. Least Absolute Shrinkage and Selection Operator) dodaje do funkcji kosztu termin proporcjonalny do sumy bezwzględnych wartości współczynników modelu. Metoda Lasso jest techniką regresji służącą do analizy danych, która może wybierać modele poprzez zmniejszenie (do zera) niektórych współczynników regresji, co jest szczególnie użyteczne, gdy mamy do czynienia z problemem wielowymiarowości.

W przeciwieństwie do regresji grzbietowej, której termin regularyzacji jest proporcjonalny do kwadratów wartości współczynników, Lasso używa sumy wartości bezwzględnych współczynników, co sprzyja tworzeniu modeli z mniejszą liczbą cech.

Parametr λ kontroluje siłę penalizacji, tak jak w regresji grzbietowej, a znalezienie odpowiedniej wartości jest kluczowe, dla osiągnięcia satysfakcjonujących wyników. Jest on z góry określonym parametrem, działa on odwrotnie do liczby zmiennych w modelu, co znaczy, że wraz prostszym modelem z mniejszą liczbą zmiennych wyznacza jest wyższą wartością λ. Istnieją różne strategie wyboru λ. Pierwszą metodą jest pośrednie podejście, gdzie przyjmuje się kryteria informacyjne, takie jak Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC) czy skorygowany współczynnik determinacji (Adjusted-R^2), oceniają jakość dopasowania modelu, penalizując jednocześnie za nadmierną złożoność. W praktyce poszukiwane są wartości λ, które minimalizują wartości AIC lub BIC, sugerując tym samym model optymalny pod kątem trade-offu między dopasowaniem, a złożonością. Kolejne są metody bezpośrednie, gdzie techniki takie jak walidacja krzyżowa (ang. cross-validation, w skrócie: CV) dzieli zestaw danych na części trenujące i testowe, a następnie model poddawany jest ocenie na niezależnym zestawie testowym. Optymalne λ jest wybierane w oparciu o wydajność modelu, na przykład, przez minimalizację średniego błędu kwadratowego na zestawie walidacyjnym. Można także użyć wiedzy eksperckiej, gdy ograniczenia interfejsu użytkownika lub wytyczne dotyczące interpretowalności, będą dyktować potrzebę utrzymania maksymalnej liczby zmiennych poniżej określonego progu. Ekspert może ręcznie dostosować wartość λ, aby uzyskać model z żądaną liczbą niezerowych współczynników, który jest jeszcze skuteczny, ale jednocześnie spełnia zewnętrzne ograniczenia. Jednak ostateczny wybór λ często wymaga iteracyjnego podejścia, łączącego powyższe strategie z wiedzą dziedzinową oraz praktycznymi wymaganiami projektu. Wzór dla regresji Lasso wygląda następująco:

(2.5)

gdzie:

= wektor estymowanych współczynników regresji uzyskanych za pomocą metody Lasso

*arg min β* = oznacza, że szukamy wartości wektora *β*, które minimalizują wyrażenie wewnątrz nawiasów

= Jest to suma kwadratów reszt (błąd regresji) podzielona przez 2, co jest standardowym terminem w regresji liniowej. Mierzy różnicę między rzeczywistymi wartościami *yi*​ a wartościami przewidywanymi przez model

= człon regularyzacyjny Lasso. Dodaje karę za sumę wartości bezwzględnych współczynników *βj*​. Parametr *λ* kontroluje siłę regularyzacji. Większe wartości *λ* prowadzą do większej redukcji wartości współczynników *βj*​, co może skutkować zerowymi współczynnikami i tym samym selekcją zmiennych

= suma kwadratów reszt z podziałem przez 2, gdzie to obserwowane wartości zmiennej zależnej, a wartości zmiennych niezależnych

Regresja Lasso ma na celu identyfikację zmiennych i odpowiadających im współczynników regresji, które prowadzą do modelu minimalizującego błąd przewidywania. Osiąga się to poprzez nałożenie ograniczenia na parametry modelu, które „zmniejsza” współczynniki regresji do zera, czyli wymusza, aby suma wartości bezwzględnych współczynników regresji była mniejsza od ustalonej wartości (λ) (Ranstam & Cook, 2018), co efektywnie eliminuje najmniej ważne cechy z modelu. Zdolność ta oznacza, że może być ona stosowana nie tylko do zapobiegania overfittingu ale także do selekcji cech modelu.

Regularyzacja L2, znana również jako regresja grzbietowa (ang. ridge), dodaje do funkcji kosztu termin proporcjonalny do sumy kwadratów współczynników modelu. Innymi słowy, karze ona za złożoność modelu.

W regresji grzbietowej, celem jest minimalizacja sumy kwadratów reszt, podobnie jak w tradycyjnej regresji liniowej, ale z dodatkowym czynnikiem regularyzującym, który zwiększa koszt posiadania dużych współczynników. Czynnik ten jest kontrolowany przez parametr λ, który jest z góry określonym parametrem. Jeśli λ jest bliskie zero, regresja grzbietowa dąży do tradycyjnej regresji liniowej; gdy λ rośnie, większą wagę przykłada się do zmniejszenia wartości współczynników. Parametr λ musi być dobrany odpowiednio, często przy użyciu technik walidacji krzyżowej, aby znaleźć optymalny balans między dopasowaniem modelu a jego zdolnością do generalizacji na nowe dane.

Wzór składa się z następujących części:

(2.6)

gdzie:

= estymator współczynników regresji grzbietowej

*arg min β* = wskazuje, że szukamy takich wartości wektora *β*, które minimalizują wyrażenie w nawiasach klamrowych

to suma po wszystkich N obserwacjach

= rzeczywiste wartości zmiennej zależnej

*β0* = wyraz wolny

= wartości zmiennych niezależnych

= suma iloczynów wartości zmiennych niezależnych i odpowiadających im współczynników regresji

= termin regularyzujący, który karze za duże wartości współczynników regresji

Regularyzacja L2 w wyniku nałożenia kary na wartości większych współczynników, umożliwia zwiększenie odporności na zjawisko nadmiernego przeuczenia. Pomimo to, w odróżnieniu do regularyzacji L1, nie prowadzi ona do sparsyfikacji modelu poprzez całkowite eliminowanie niektórych współczynników, lecz zachowuje ich wartość, choć pomniejszoną.

Dla klasyfikatorach uczenia głębokiego sprawdzono także dodanie kroku: „Dropout”, który jest stosowany w sieciach neuronowych w celu zapobiegania nadmiernemu dopasowaniu. Technika ta została wprowadzona przez Srivastavę i współpracowników w 2014 roku i szybko stała się popularnym elementem w projektowaniu głębokich sieci neuronowych.

Podczas treningu, metoda dropout losuje z ekspotencjalnie dużej liczby różnych "przyciętych" sieci. W czasie testowania łatwo jest przybliżyć efekt uśredniania przewidywań wszystkich tych przyciętych sieci, po prostu używając jednej pełnej sieci o mniejszych wagach (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). Działanie tego algorytmu polega na losowym wykluczaniu (poprzez ustawnie wartości zero) części neuronów w warstwach sieci z określonym prawdopodobieństwem *p.* Schemat działania metody dropout został pokazany na rysunku 22, gdzie pokazano różnicę między konwencjonalną siecią neuronową i po zastosowaniu opisywanego podejścia. Pełna sieć z wszystkimi aktywnymi neuronami, ma połączenia między wszystkimi możliwymi krawędziami oraz warstwami.

W sieci neuronowej po zastosowaniu dropout ma częściowo dezaktywowane neurony (oznaczone jako przekreślone), co oznacza, że te neurony oraz połączenia między nimi zostały tymczasowo wyeliminowane z procesu treningowego. Gdy neuron jest wykluczony, niesie to za sobą odpowiedzialność w postaci braku przekazywania informacji do przodu przez te jednostki w sieci i w ten sposób nie uczestniczą w procesie propagacji wstecznej. Wskazane wcześniej prawdopodobieństwo *p* jest hiperparametrem modelu i jest ono dobierane w zależności od potrzeb.



Rysunek 22. Zastosowanie metody dropout w konwencjonalnej sieci neuronowej. Źródło: opracowanie własne

Losowe „wyłączanie” neuronów pozwala na inny odbiór, choć dane wejściowe pozostają te same, sposób, w jaki model na nie reaguje i z nich uczy się, ulega zmianie. To z kolei zmusza model do rozwijania bardziej uogólnionych i uniwersalnych strategii przetwarzania danych, poprawiając jego zdolność generalizacji. Najszerzej wykorzystuje się to podejście w gęsto połączonych warstwach sieci neuronowych, jednak w zależności od specyfiki problemu i architektury cechującej model dropout ma także swoje zastosowanie w warstwach konwolucyjnych i rekurencyjnych. Podsumowując, technika dropout to szeroko stosowana technika regularyzacji, często niezbędna do osiągnięcia najnowocześniejszych wyników dla wielu architektur (Wei, Kakade, & Ma, 2020). Poprzez losowe dezaktywowanie części neuronów podczas treningu, dropout zachęca sieć do uczenia się bardziej odpornych i ogólnych cech, co ostatecznie prowadzi do poprawy wydajności na nieznanych danych testowych. Ta zdolność zwiększa uogólnienie modelu i skutecznie łagodzi ryzyko przeuczenia, co skutkuje tworzeniem bardziej solidnych i niezawodnych sieci (Salehin & Kang, 2023).

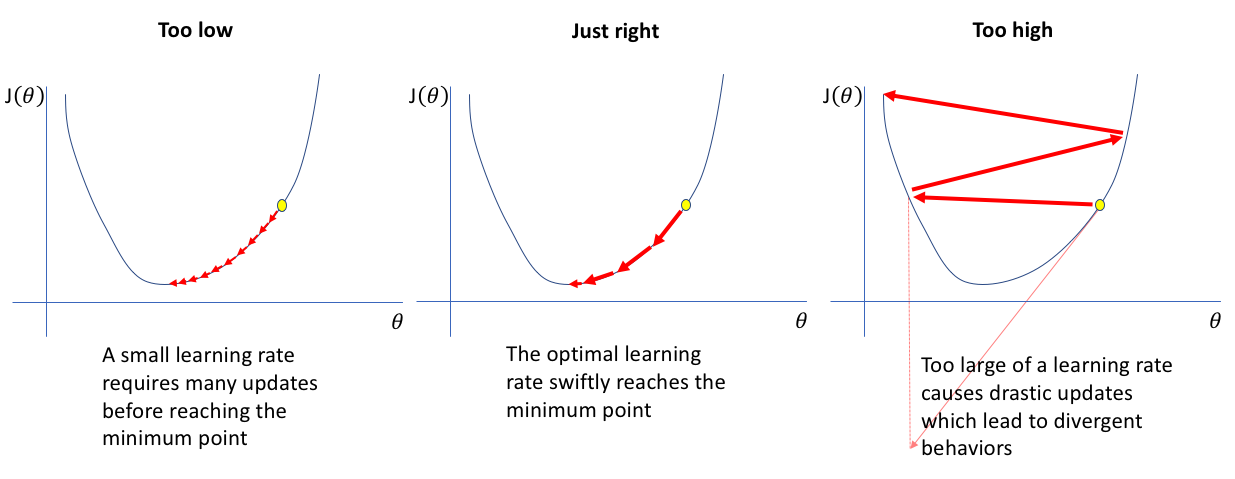
## 2.5 Optymalizacja

Regularyzacja pomaga w zapobieganiu zjawiska nadmiernego dopasowania, ale w procesie budowania skutecznych modeli uczenia maszynowego kluczowe jest także dobranie optymalnych wartości parametrów modelu, aby maksymalizować jego wydajność dla danych testowych, a za to odpowiada optymalizacja.

Uczenie maszynowe rozwija się dynamicznie, co doprowadziło do wielu przełomów teoretycznych i znalazło szerokie zastosowanie w różnych dziedzinach. Optymalizacja jako istotna część uczenia maszynowego, przyciąga dużą uwagę badaczy (Sun, Cao, Zhu, & Zhao, 2019). Istnieją głównie dwa rodzaje metod optymalizacji hiperparametrów: ręczne i automatyczne.

Ręczne poszukiwanie polega na próbnym dobieraniu zestawów hiperparametrów manualnie. Zależy ono od podstawowej intuicji i doświadczenia ekspertów, którzy potrafią zidentyfikować ważne parametry mające większy wpływ na wyniki, a następnie określić związek między pewnymi parametrami, a ostatecznymi wynikami za pomocą narzędzi wizualizacji (Wu, Chen, Zhang, Xiong, Lei, & Deng, 2019). Optymalizacja (ang. optimization) w świcie uczenia maszynowego jako swój główny cel wyznacza określenie procesu dostosowania parametrów i struktury budowanego modelu w celu zminimalizowania funkcji kosztu. Zmniejszenie wartości tej funkcji jest podstawowym zagadnieniem, ponieważ prowadzi do stworzenia modelu, który lepiej spełnia warunki zadania, a może to zostać osiągnięte poprzez dostosowanie przez model swoich parametrów, takich jak wagi w sieciach neuronowych, czy współczynnik nachylenia w regresji liniowej. Model, który będzie posiadał pomniejszoną funkcję kosztu, będzie dokonywał dokładniejszych predykcji, będzie posiadał umiejętność generalizacji i przez to będzie pracował efektywniej na danych, których wcześniej nie badał, a w ten sposób będzie można zapobiegać przeuczeniu.

Bezpośrednio z funkcją kosztu związany jest współczynnik uczenia (ang. learning rate), który jest krytycznym parametrem określającym wielkość kroku podczas aktualizacji działania kolejnych iteracji. W sytuacji, gdy współczynnik uczenia jest mały, kroki wykonywane będą miały małą wielkość, a działanie algorytmu w celu znalezienia minimum będzie czasochłonne (rysunek 23). Z drugiej strony, dobranie współczynnika uczenia, o zbyt dużej wartości może spowodować, że model „przeskoczy” minimum. Taka sytuacja będzie prowadzić do niestabilnych i oscylujących aktualizacji, a model może stać się coraz mniej dokładny. Celem badacza jest zawsze znalezienie optymalnej wartości współczynnika uczenia, aby w sposób możliwie najszybszy zbliżać się do minimum, osiągając najlepsze wyniki. Odnalezienie odpowiedniego współczynnika uczenia jest często przedmiotem długotrwałej pracy eksperckiej, doświadczenia, ale także może wymagać eksperymentów podczas kolejnych iteracji.



Rysunek 23. Porównanie małego, optymalnego i dużego współczynnika uczenia. Źródło: Jordan (2018)

Istnieje wiele sposobów, aby osiągnąć ten cel. Jednym z nich jest optymalizacja symulacyjna, którą można zdefiniować jako proces znajdowania najlepszych wartości zmiennych wejściowych spośród wszystkich możliwości bez jawnego oceniania każdej z możliwości (Carson & Maria, 1997). Pierwszym z nich będzie zdefiniowanie odpowiedniej funkcji kosztu, ponieważ podejścia mogą się różnić w zależności od natury zadania. Warto także eksperymentować z różnymi wartościami hiperparametrów, takimi jak na przykład liczba epok. Można także użyć walidacji krzyżowej lub zbioru walidacyjnego, aby ocenić wydajność modelu, a co za tym idzie uniknąć przeuczenia. Osiągnąć ten cel można także przez wykorzystanie regularyzacji, zwłaszcza gdy dane charakteryzują się z dużą ilością cech lub małą ilością doświadczeń. Należy również zadbać o podstawowe aspekty, takie jak przygotowanie danych (np. standaryzacji, przetworzenie braków danych), monitorowanie pracy modelu czy zwrócenie uwagi na posiadane zasoby sprzętu komputerowego.

Wszystkie te aspekty są niebywale istotne w procesie tworzenia modelu spełniającego warunki zadania, ale w wielu przypadkach kluczowe będzie znalezienie odpowiedniego algorytmu optymalizacji. Te metody numeryczne umożliwiają algorytmowi na wykrywanie schematów z danych i poprawę wcześniej wyznaczonych prognozy. Jako popularne algorytmy optymalizacji uważa się: Stochastic Gradient Descent (SGD), AdaGrad, Momentum, RMSProp oraz Adam Optimizer. Stochastic Gradient Descent to podstawowa forma algorytmu opartego na gradientach. Jego stochastyczność (ang. stochastic), oznacza, iż jako dane wejściowe nie bierze on całego zestawu danych, lecz losowo wybrany zestaw, w celu wyznaczenia funkcji kosztu. W tej metodzie parametry modyfikowane są często. Aby to zwizualizować, można przyjąć, że zbiór danych posiadanych ma 10 wierszy, co oznacza, że SGD zaktualizuje parametry modelu 10 razy w jednym cyklu, zamiast zrobić to tylko jeden raz, jak to ma miejsce w standardowym spadku gradientowym (ang. Gradient Descent). Działanie to ma swoje następstwa w wysokiej wariancji, fluktuacjach funkcji straty o wielorakich intensywnościach. AdaGrad aktualizuje wielkość kroku w SGD na bieżąco, zgodnie z gradientami otrzymanymi w trakcie procesu; metody te zdobyły szerokie zastosowanie w optymalizacji na dużą skalę ze względu na swoją zdolność do robustnego zbiegania, bez potrzeby precyzyjnego dostosowywania harmonogramu wielkości kroku (Ward, Wu, & Bottou, 2020).

Momentum jest to modyfikacją algorytmu SGD, ma on umiejętność zbierania kierunków uprzednich kroków, w celu osiągnięcia płynniejszego i szybszego zbieganie, w ten sposób ułatwia ustawienie odpowiedniego kierunku przemieszczania się algorytmu, a w ten sposób zmniejsza oscylację oraz przyspiesza zbieżność.

RMSProp jest modyfikacją algorytmu AdaGrad, w swoim zamyśle zwraca uwagę tylko na znak gradientu, nie na jego wielkość, a także adaptowane są indywidualnie rozmiary kroków dla poszczególnych wag, co daje możliwość na naukę w efektywniejszy sposób nawet w obecności płaskich obszarów i punktów siodłowych. Kolejny krok w zależności od poprzednich gradientów jest mnożnikowo zwiększany lub zmniejszany.

Algorytm optymalizacji Adam (ang. Adaptive Moment Estimation) łączy zalety algorytmów Momentum i RMSProp. Nie tylko gromadzi historyczne gradienty i stopniowo zmniejsza tempo uczenia się, jak RMSProp, ale również rejestruje wartość wykładniczego zaniku gradientu, podobnie jak algorytm Momentum (Yang & Long, 2023).

Podsumowując, znaczenie optymalizacji polega nie na próbie poznania wszystkiego o systemie, ale na znalezieniu, przy jak najmniejszym wysiłku, najlepszego sposobu dostosowania systemu. Jeśli jest to dobrze wykonane, systemy mogą mieć bardziej ekonomiczny i ulepszony projekt, mogą działać bardziej precyzyjnie lub za niższe koszty, a projektant systemu będzie miał lepsze zrozumienie wpływu interakcji i zmienności parametrów na jego projekt (Adby & Dempster, 1974).

## 2.6 Metody redukcji wymiarów

W analizie danych tekstowych, szczególnie, gdy zbiór danych ma wiele obserwacji i jest złożony, często występuje problem wysokiej wymiarowości. Redukcja wymiarowości to często etap mający duży wpływ na sukces stworzenia prawidłowego modelu uczenia maszynowego. Pomaga to nie tylko złagodzić klątwę wymiarowości, ale może także prowadzić do znacznej poprawy interpretowalności wyników, a także wydajności obliczeniowej. Redukując wymiarowość, kompresujemy informacje w sposób zachowujący najbardziej odpowiednie struktury istotne dla danego zadania (Bishop, 2011). Każde słowo w zestawie danych może być traktowane jako osobna cecha, a przez to wymiar. Taka sytuacja w sposób znaczący zwiększa złożoność obliczeniową i będzie potrzebować więcej pamięci do przetwarzania, jak również będzie obniżać skuteczność algorytmów uczenia maszynowego, czy zmniejszać ryzyko nadmiernego dopasowania, przez zmianę na lepsze generalizacji. Kolejnym aspektem jest ułatwienie wizualizacji danych przy niższej wymiarowości, a przez to uproszczenie interpretacji. W pracy używane są algorytmy PCA i Truncated SVDdo redukcji wymiarów i przekształcania danych.

Analiza głównych składowych (ang. Principal Component Analysis, w skrócie: PCA) jest techniką redukcji wymiarów, która przekształca zbiór zmiennych zależnych od siebie w zbiór nowych, niezależnych zmiennych. Celem algorytmu PCA jest wydobycie kluczowych informacji z danych statystycznych w celu przedstawienia ich jako zestawu nowych zmiennych ortogonalnych, zwanych składowymi głównymi oraz pokazanie wzorca podobieństwa między obserwacjami i zmiennymi (Kurita, 2021). Główne składowe to kilka liniowych kombinacji oryginalnych zmiennych, które maksymalnie wyjaśniają wariancję wszystkich zmiennych. W procesie tym, ta metoda zapewnia przybliżenie oryginalnej tabeli danych, wykorzystując jedynie te kilka głównych składowych (Greenacre, Groenen, Hastie, & D'Enza, 2022). Dzięki temu można uprościć dane, zachowując najważniejsze informacje.

Truncated Singular Value Decomposition (w skrócie: Truncated SVD) jest techniką podobną do PCA, ale jest bardziej odpowiednia dla rzadkich macierzy, takich jak macierze dokumentów w analizie tekstu. Badania sugerują, że zredukowana SVD (od ang. Singular Value Decomposition) jest korzystną alternatywą dla standardowej regularyzacji w przypadkach macierzy źle uwarunkowanych, ale z dobrze określoną rangą numeryczną. (Hansen, 1987). W problemach liniowych najmniejszych kwadratów, które są źle postawione, standardowe metody mogą nie dawać stabilnych wyników. Zredukowana SVD, polegająca na odcięciu najmniejszych wartości osobliwych, pomaga w uzyskaniu stabilniejszych rozwiązań. W przeciwieństwie do PCA, ten estymator nie centruje danych przed obliczeniem rozkładu wartości osobliwych (scikit-learn.org, 2024). W PCA dane są standaryzowane (centrowane), co oznacza, że średnia każdej zmiennej jest przesuwana do zera przed obliczeniami. Natomiast w Truncated SVD dane nie są centrowane, co sprawia, że metoda ta jest bardziej odpowiednia dla danych rzadkich i dużych macierzy, na przykład w analizie tekstu. Metoda TSVD sprowadza się do obcięcia rozszerzenia wartości osobliwych macierzy współczynników (Hansen, Sekii & Shibahashi, 1992). Wartości osobliwe są to wartości określające rozciągłość macierzy w różnych kierunkach, dzięki odrzuceniu najmniejszych wartości osobliwych, które mogą odpowiadać za szum lub niestabilność, można uzyskać racjonalne wyniki nawet przy źle uwarunkowanych macierzach. Wartości osobliwe są numerycznymi wartościami, które charakteryzują rozkład macierzy na jej elementarne składowe, które określają, jak każdy wymiar danych przyczynia się do ogólnej struktury danych (Zhang, 2023).

Podsumowując w metodzie Truncated SVD redukuje się wymiary poprzez odrzucenie tych składowych, które mają najmniejsze wartości osobliwe. Metoda ta jest wykorzystywana w zadaniach kompresji danych, redukcji szumu danych. W programie stworzonym na potrzeby tej pracy użyto zdefiniowanej funkcji TruncatedSVD() z biblioteki sklearn.decomposition.

# 3 Analiza Sentymentu w mediach społecznościowych

Poniższy rozdział został poświęcony opisowi danych, metod, procedur wykorzystanych w projekcie analizy sentymentu danych z mediów społecznościowych. Elaboruje on przyczyny selekcji metod w ramach wybranych modeli uczenia maszynowego oraz dostarcza sukcesywnego opisu ich zastosowania w kolejnych etapach. Zostaje również dostarczony szczegółowy wgląd w dane, poczynając od źródła pozyskania, poprzez eksploracyjną analizę danych, aż po techniki ich przetwarzania wstępnego potrzebne do ostatecznego zasilenia modeli. Celem badania jest analiza i klasyfikacja sentymentów wyrażanych w zdaniach pochodzących z blogów udostępnionych przez *Opinmind*. W ostatecznej części określone są wymagania technologiczne, potrzebne do uruchomienia projektu.

## 3.1 Charakterystyka i przygotowanie danych

Dane, które będą służyły jako treningowe i testowe pochodzą ze strony kaggle.com, która jest platformą Internetową dedykowaną dla osób ze środowiska Data Science. *Kaggle* umożliwia użytkownikom wyszukiwanie i publikowanie zbiorów danych, eksplorowanie i budowanie modeli w internetowym środowisku nauki o danych, współpracę z innymi badaczami danych i inżynierami uczenia maszynowego oraz udział w konkursach w celu rozwiązywania wyzwań związanych z nauką o danych (wikipedia.org, 2024).

Dane zostały pobrane w ramach konkursu zorganizowanego przez *University of Michigan SI650* (Information Retrieval) „UMICH SI650 - Sentiment Classification” i zostały one zebrane z mediów społecznościowych, pierwotnie z witryny opinmind.com (analyticsindiamag.com, 2024), co daje możliwość analizy danych bezpośrednio zebranych od autentycznych użytkowników, korzystających z sieci społecznościowych. Zadaniem tego konkursu jest klasyfikacja sentymentu, poprzez określenie czy dana wypowiedź wyraża pozytywne czy negatywne emocje, co spotyka cel tej pracy.

W projekcie wykorzystywane są dane ówcześnie podzielone na dwa zbiory: treningowy, testowy i pierwotnie podane w formacie tekstowym, gdzie każda linia reprezentuje jeden wpis z mediów społecznościowych. Dane treningowe zawierają 7086 zdań, każde już oznaczone etykietą 1 (sentyment pozytywny) lub 0 (sentyment negatywny). Dane testowe składają się z 33052 zdań, które nie mają przypisanych etykiet i które uczestnicy konkursu muszą samodzielnie zaklasyfikować (kaggle.com, 2024).

Oba te zbiory zostały wczytane do środowiska Python przy pomocy biblioteki *Pandas*, co umożliwiło efektywne manipulacje oraz przetwarzanie. Biblioteka ta pozwala na przygotowanie ramki danych (ang. data frame), która następnie jest używana w aplikacji. Z pliku tekstowego została stworzona reprezentacja tabelaryczna - kolumny, reprezentujące sentyment i treść komentarza, oraz zostały obsłużone tabulacje, cudzysłowy oraz brak nagłówków w pierwszej linii.

Następnie przeprowadzono przygotowanie danych, gdzie każdy z kroków ma na celu uproszczenie i optymalizację danych wejściowych. Proces ten ma zwiększyć dokładność, efektywność, ale w problemie tym problemie, przede wszystkim umożliwić klasyfikację sentymentu dokumentów. Przygotowanie danych to nie tylko pierwszy krok, ale należy go powtarzać wielokrotnie w trakcie analizy, gdy dowiadujemy się więcej o danych i stosowanych przez nas technikach modelowania (Provost & Fawcett, 2013)

Jako pierwszy krok dokonano analizy pobranych danych, aby zrozumieć charakterystykę i móc prawidłowo zdiagnozować jakie kolejne etapy musza być zaimplementowane. „Garbage in, garbage out” - jeśli dane szkoleniowe zaimplementowanego systemu zawierają wiele braków danych, wartości odstających czy szumu (np. z powodu pomiarów o niskiej jakości), system nie może nauczyć się wykrywania schematów w stopniu zadowalającym (Géron, 2019). W tej części skupiono się na graficznej prezentacji zgromadzonych informacji, co pozwoli nie tylko na lepsze zrozumienie, ale również na wielopoziomową, dogłębną analizę, wraz z wykryciem pierwotnie niewidocznych wzorców czy zależności. Wizualizacja danych jest o tyle ważna, że umożliwia przekazanie informacji w sposób intuicyjny.

Rozkład sentymentów w danych treningowych został zwizualizowany poniżej (rysunek 24) i ilustruje liczbę wystąpień dla każdego z sentymentów, a są one oznaczone jako 0 dla negatywnych i 1 dla pozytywnych.

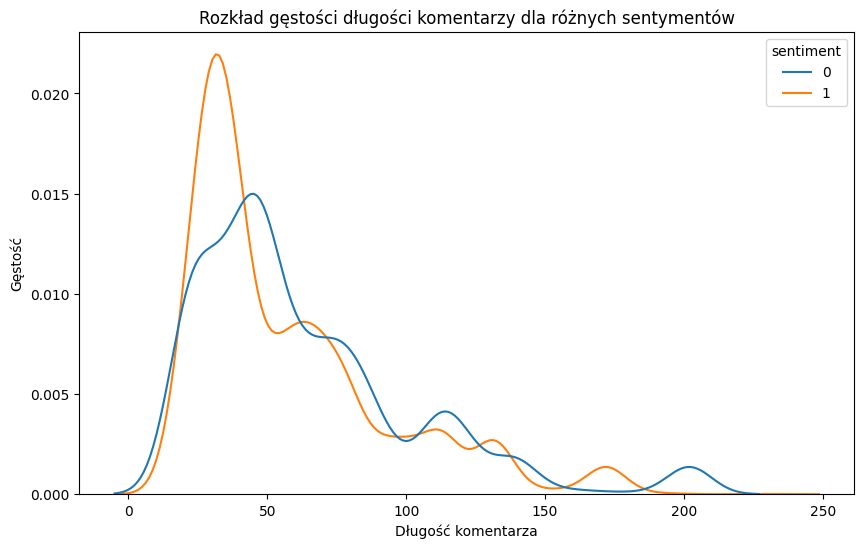
A graph with numbers and a bar

Description automatically generated

Rysunek 24. Rozkład sentymentów w danych treningowych. Źródło: opracowanie własne

Wykres ten wskazuje na brak równomiernego rozłożenia między klasami. W idealnej sytuacji, aby zapewnić najmniejszą stronniczość modelu, najlepiej posiadać dane, gdzie różnica ta jest minimalna. Na podstawie wiedzy eksperckiej określono, że różnica pomiędzy liczbą instancji w klasach nie jest znacząca, co może oznaczać, że klasyfikator będzie mógł nauczyć się efektywnie rozpoznawać obie klasy bez interwencji w rozkład danych, zwrócono jednak uwagę na dobranie odpowiednich metryk oceny, które lepiej odzwierciedlają efektywność modelu w warunkach niezbalansowanych klas. Zamiast arbitralnego ponownego próbkowania danych, w celu poradzenia sobie z brakiem równowagi, zastosowano wrażliwe na koszty wskaźniki uczenia się i oceny, takie jak pole pod krzywą ROC (He & Ma 2013). Współczesne algorytmy uczenia maszynowego, zwłaszcza te oparte na głębokim uczeniu, są często bardzo odporne na problem niezbalansowanych danych i mogą być efektywne nawet bez konieczności stosowania *samplingu*. Warto także zauważyć, że nie dokonano manipulowania rozkładem, aby stworzyć zestaw danych idealnych, ponieważ może to prowadzić do nadmiernego dopasowania, zmniejszyć zdolność generalizacji, dlatego istotne jest, aby dać możliwość pracy modelowi na danych zbliżonych do naturalnych, takich które mogą występować w świecie rzeczywistym. Niezwykle istotne jest ostrożne stosowanie technik samplingu i zapewnienie, że metoda walidacji dokładnie odzwierciedla działanie modelu na nowych, niewidocznych danych (He & Garcia, 2009). Techniki takie jak dostosowywanie progu decyzyjnego i stosowanie metod wykrywania anomalii mogą być bardziej skuteczne niż próbkowanie w niektórych kontekstach, zwłaszcza gdy klasa mniejszości jest bardzo mała (Aggarwal, 2014).

Następnym krokiem w analizie jest stworzenie wizualizacji rozkładu gęstości długości, dla sentymentu 1 i dla sentymentu 0 (rysunek 25), która przedstawia współczynnik korelacji dwupunktowej, który jest odpowiedni dla zmiennej binarnej: *sentiment* (reprezentującą klasę sentymentu) oraz zmiennej ciągłej: *comment\_length* (określającą długość komentarza).



Rysunek 25. Rozkład gęstości długości komentarzy dla różnych sentymentów. Źródło: opracowanie własne

Istnieje minimalna tendencja do tego, że dłuższe komentarze mogą charakteryzować się bardziej negatywnym nastrojem. Jednakże, stwierdza się, że długość komentarza nie jest optymalnym wskaźnikiem dla przewidywania jego sentymentu. Komentarze dłuższe i krótsze są niemal równie prawdopodobne, aby być pozytywne lub negatywne. Wynik ten wskazuje, że inne cechy komentarzy będą bardziej decydujące dla określenia sentymentu wpisu. Długość komentarza, w tym przypadku, nie dostarcza statystycznie istotnych informacji na temat jego sentymentu.

Warto także pamiętać, że złożoność zdania zależy od wielu czynników oprócz długości: znajomości słów, abstrakcyjności słów, automatyzacji wzorców słów używanych w zdaniu itp. (Mobayyen & de Almeida 2005).

Następnie w pracy stworzono grafikę, który reprezentuje najczęściej występujące terminy w pozytywnych komentarzach ze zbioru treningowego (rysunek 26). Dokonano tego za pomocą chmury słów, która ukazała, że wyraźnie dominującym tematem są nazwy szeroko obecnych w pop kulturze książek i filmów, takich jak „*Harry Potter*” czy „*Mission Impossible*”.

A close up of words

Description automatically generated

Rysunek 26. Chmura słów dla terminów występujących w pozytywnych komentarza zestawu danych treningowych. Źródło: opracowanie własne

Wizualizacja ta pozwala zidentyfikować, główne tematy komentarzy. Poza tematami dzieł kulturalnych, pojawiają się często słowa opisujące emocje, takie jak „*love*”, „*great*”, co upewnia, że dokumenty o pozytywnym sentymencie, charakteryzują się ogólnie pozytywnym tonem. Opinie posiadają też słowa o zabarwieniu osobistym, takie jak „*loved*”, „*liked*”, określające personalne odczucia autora. W kontekście analizy sentymentu chmura słów pozwoliła na natychmiastową identyfikację dominujących trendów, natomiast wykonano także wykres słupkowy przedstawiający częstotliwość występowania dziesięciu najczęściej używanych słów w pozytywnych jak i negatywnych komentarzach (rysunek 27).

A graph of different colored bars

Description automatically generated

Rysunek 27. Częstotliwość występowania 10 najczęściej używanych słów w pozytywnych i negatywnych komentarzach. Źródło: opracowanie własne

Dominującymi słowami w pozytywnych komentarzach są słowa: "*love*", "*awesome*", "*Harry*", "*Potter*" i "*code*". Wskazuje to, że użytkownicy często komunikują swoje zadowolenie w kontekście specyficznych wyrażeń i zainteresowań takich jak filmy („*Harry Potter*”, czy „*The Da Vinci Code*”). W negatywnych komentarzach najczęściej występują słowa nacechowane pejoratywnie, takie jak „*hate*”, „*sucks*”. Terminy te są bezpośrednim przejawem niezadowolenia i krytyki. Warto zauważyć, że występują również słowa neutralne, pojawiające się zarówno w komentarzach pozytywnych jak i negatywnych. Jest to przejaw złożoności problemu analizy sentymentu i pokazanie jak ważne jest badanie kontekstu a nie tylko pojedynczych słów.

Kolejnym etapem przetwarzania danych jest podział na zestaw treningowy i walidacyjny. Chociaż w projekcie zbiór danych został podzielony na samym początku na treningowy i testowy, to ponowny podział na zestaw danych treningowych i walidacyjnych ma zasadnicze znaczenie dla skutecznego trenowania i weryfikowania modeli (rysunek 28). Pierwotny zestaw danych testowych nie zawiera zdefiniowanych etykiet (czyli kolumny/ zmiennej „*sentiment*”), a walidacyjny zbiór będzie ją zawierał. Zdefiniowana zmienna celu umożliwia ocenę jak dobrze wytrenowany model działa na nowych danych, czyli jego zdolność predykcji. Niezbędne jest posiadanie osobnego zestawu danych – zwanego zestawem deweloperskim (programistycznym) lub zbiorem walidacyjnym – na podstawie którego ocenia się wybrany algorytm. Możesz dostroić swój model na zestawie szkoleniowym, ale to zestaw deweloperski dostarcza informacji zwrotnych potrzebnych do poprawy uogólnienia modelu (Ng, 2018). Powszechnym wyborem jest wykorzystanie 70% danych do uczenia, 10% do walidacji, a pozostałe 20% do testowania. Współczynniki te można dostosować w zależności od ilości dostępnych danych i specyficznych wymagań aplikacji (Bishop, 2011). Zbiór treningowy wykorzystywany jest do budowy, trenowania modelu, estymowania kandydatów, stanowi on 60 - 70% podzbiór całego zbioru danych. Zbiór walidacyjny służy do wstępnej oceny modeli i doboru hiperparametrów, celem jego stworzenia jest wykorzystanie go w wyborze jednego najlepszego modelu, przyjmuje się zazwyczaj 15 - 20% danych jako ten zestaw. Zbiór testowy umożliwia realizację ostatecznej oceny modelu, przez nieobciążoną estymację błędów ostatecznego, jednego wybranego modelu. Najczęściej tworzy się zbiór testowy jako ostatnie 15 - 20% posiadanych danych. Zbiór walidacyjny służy również do doboru hiperparametrów modelu, poprzez eksperymentowanie z różnymi ustawieniami, aby znaleźć model optymalny. Dzięki zastosowaniu oddzielnego zestawu walidacyjnego do dostrojenia hiperparametrów model może lepiej uogólniać, ponieważ nie został bezpośrednio zoptymalizowany pod kątem wydajności wyłącznie na danych szkoleniowych (James, Witten, Hastie & Tibshirani, 2013). Bez zestawu testowego istnieje ryzyko nadmiernego dopasowania do danych szkoleniowych, co spowodowałoby niepowodzenie modelu w przypadku nowych, niewidocznych danych (Géron, 2019).

A diagram of a diagram

Description automatically generated

Rysunek 28. Wybór optymalnego modelu za pomocą podziału danych na trzy zbiory. Źródło: opracowanie własne

W praktyce podział na zbiór treningowy i walidacyjny można implementować na wiele sposobów, w zależności od potrzeb, dostępności, specyfiki problemu oraz wiedzy eksperckiej. W aplikacji został dokonany podział za pomocą funkcji „*train\_test\_split()*” z biblioteki *scikit-learn*, używając proporcji 70/30. Taki podział zapewnia wystarczająco duży zbiór danych do trenowania, aby modele mogły nauczyć się zróżnicowanych wzorców, ale w tym samym czasie tworzony jest zbiór testowy umożliwiający efektywne ocenianie modeli.

Następnie przeprowadzono analizę przy użyciu technik TF (Term Frequency) i TF-IDF (Term Frequency-Inverse Document Frequency), a poprzedzało ją przetwarzanie każdego zdania na listę ujednoliconych słów poprzez stemming i tokenizację. Stemming redukuje wielorakie formy podanego słowa do podstawowej. Natomiast tokenizacja to mechanizm polegający na podzieleniu analizowanego tekstu na tzw. tokeny, które są mniejszymi jednostkami. TF przekształca zadany tekst na macierz cech, gdzie każda kolumna reprezentuje słowo, każda komórka zawiera liczbę wystąpień w dokumencie (tabela 2). Macierz ta pozwala na bezpośrednie wykorzystanie danych w modelach klasyfikacyjnych.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | a | about | an | and | angelina | are | as | at | awesom | ... | want | way | we | when | which |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | … | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 |

Tabela 2. Macierz cech po transformacji Term Frequency. Źródło: opracowanie własne

TF pozwala na uzyskanie liczbowej reprezentacji dla częstości słów, co jest przydatne w podstawowej analizie sentymentu.

A close up of words

Description automatically generated

Rysunek 29. Wizualizacja dominujących terminów w opiniach. Źródło: opracowanie własne

Za pomocą *CountVectorizer* została stworzona macierz terminów, gdzie każdy wiesz odpowiada komentarzowi (tabela 2), a kolumny odpowiadają słowom występującym z danych. Aby zwizualizować jak często występuje każde słowo użyto chmury słów (rysunek 29), gdzie słowa są skalowane proporcjonalnie do jego częstości. Grafika pokazuje, że w zbiorze jest dużo wyrazów, które nie wpływają na sentyment takie jak: „*the*”, „*of*”, a są one dominujące. Widocznie często występują też opisy emocji: „*love*”, „*hate*” lub określenia wskazujące na kontekst komunikatu, specyfikę rozmowy, typu: „*Honda*” czy „*London*”.

Natomiast macierz TF-IDF zawiera wartości (tabela 3), które są proporcjonalne do liczby wystąpień słowa w dokumencie, jednocześnie uwzględniając liczbę dokumentów – ten sposób transformacji słowa rzadko występujące uzyskują większe wagi, w tym samym czasie słowa często występujące o małym znaczeniu będą miały ograniczone znacznie. Do tej transformacji tekstu w projekcie użyto *TfidfVectorizer* z pakietu *sklearn*.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | a | about | an | and | angelina | are | as | at | awesom | ... | want | way | we | when | which |
| 0 | 0.000 | 0.0 | 0.0 | 0.0 | 0.000 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.000 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1 | 0.000 | 0.0 | 0.0 | 0.0 | 0.169 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.229 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 0.325 | 0.0 | 0.0 | 0.0 | 0.000 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.000 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3 | 0.325 | 0.0 | 0.0 | 0.0 | 0.000 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.000 | 0.0 | 0.0 | 0.0 | 0.0 |
| 4 | 0.000 | 0.0 | 0.0 | 0.0 | 0.000 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.000 | 0.0 | 0.0 | 0.0 | 0.0 |

Tabela 3. Macierz cech po transformacji Term Frequency-Inverse Document Frequency. Źródło: opracowanie własne

TD-IDF jest metodą, która dodaje do TF miarę odwrotnej częstotliwości dokumentów co daje możliwość reprezentacji tekstowej w sposób uwzględniający znacznie więcej czynników, niż tylko częstotliwość występowania wyrazu, ale także realizuje wprowadzenie elementu stopnia istotności danego słowa. W ten sposób często TF-IDF daje bardziej zadawalające wyniki klasyfikacji niż TF. Jednak obie metody predysponują do wykonania modelowania poprzez stworzenie reprezentacji cech zrozumiałych dla algorytmu.

## 3.2 Analiza sentymentu z użyciem narzędzi VADER i NLTK

W kontekście analizy sentymentu szczególne znaczenie mają techniki, które pozwalają na skuteczniejszą i mniej intensywnie eksploatująca zasoby interpretację emocji. W niniejszej części skupiono się na dwóch zaawansowanych narzędziach NLP: VADER (pełna nazwa: Valence Aware Dictionary and sEntiment Reasoner) i NLTK (pełna nazwa: Natural Language Toolkit), które są wykorzystywane do analizy sentymentów.

VADER to narzędzie do analizy nastrojów, oparte na leksykonie i regułach, które jest specjalnie dostosowane do nastrojów wyrażanych w mediach społecznościowych (pypi.org, 2024). Jest to projekt w pełni open-source i napisane w języku Python. Stosując kombinację metod jakościowych i ilościowych, konstruujemy i empirycznie zatwierdzamy złotą listę cech leksykalnych (wraz z powiązanymi z nimi miarami intensywności nastrojów), które są specjalnie dostosowane do nastrojów w kontekstach przypominających mikroblog (Hutto & Gilber, 2014). VADER wykazuje szczególną skuteczność w rozpoznawaniu subtelnego sposobu wyrażania nastroju. Emocje takie jak sarkazm czy ironia są często spotykane w komunikacji internetowej, a właśnie w takich formach VADER wykazuje najlepsze wyniki. Wykorzystuje on listę słów wraz z przypisaną intensywnością przekazywanego nastroju, a dodatkowo dodaje zestaw reguły gramatyczne, aby skuteczniej zaklasyfikować sentyment. VADER został wykorzystany ze względu na swoją specjalizację w analizie sentymentu krótkich tekstów, często spotykanych w mediach społecznościowych. VADER efektywnie radzi sobie z interpretacją subtelnych wyrazów nastroju, dzięki zastosowaniu modyfikatorów intensywności, unikalnego słownika z ocenami sentymentu, jak i algorytmów rozumiejących kontekst. To pozwala na bardziej zróżnicowane i dokładne rozpoznawanie nastrojów, co jest istotne w przypadku analizy opinii, reakcji czy komentarzy w mediach społecznych.

Następnym narzędziem do analizy tekstu jest NLTK, które jest wiodącą platformą do tworzenia programów w języku Python do pracy z danymi w języku ludzkim (nltk.org, 2024). Oferuje ona zarówno gotowe narzędzia (modele do analizy sentymentu), ale także ułatwia tworzenie własnych rozwiązań. NLTK jest szeroko stosowany ze względu na swoje bogate funkcjonalności w zakresie tokenizacji, normalizacji oraz eliminacji słów nieistotnych, co jest kluczowe w przygotowywaniu danych tekstowych do dalszej analizy. Umożliwia ono rozkładanie tekstu na poszczególne słowa, redukcję słów do ich podstawowych form, co pomaga w minimalizacji różnorodności lingwistycznej i skupieniu na kluczowych terminach wpływających na sentyment. Duże grono społeczności, liczba dostępnych zasobów, łatwa dostępność, modułowa budowa propaguje jego użycie wśród naukowców i praktyków NLP.

Narzędzie NLTK i VADER, zapewniają komplementarne podejście do przetwarzania i analizy sentymentu, i to właśnie dlatego zostały wybrane do użycia w aplikacji. NLTK jest instrumentem, które głównie umożliwia wstępne przetwarzania tekstu i w ten sposób otwiera drzwi do głębokiej analizy sentymentu przez stworzenie odpowiedniej struktury danych, natomiast VADER dostarcza efektywnych metod oceny sentymentu, szczególnie przydatnych do analizy informacji ze środowiska o dynamicznej charakterystyce jakim jest Internet. Dzięki zastosowanej kombinacji tych rozwiązań, aplikacja jest w stanie oferować wyniki, które mogą być wykorzystane do szerokiej gamy zastosowań badawczych tudzież biznesowych

## 3.3 Budowa i optymalizacja modeli klasyfikacyjnych

W ramach realizacji tej pracy magisterskiej skupiono się na zbudowaniu złożonych klasyfikatorów, które wykorzystują metodykę pipeline w celu efektywnego przetwarzania danych i automatyzacji. Implementacja klasyfikatorów w potoku „make\_pipeline” z biblioteki sklearn oraz połączenie tego rozwiązania z GridSearch z biblioteki scikit-learn to kluczowe podejście do zapewnienia wysokiej jakości, skuteczności modelowania.

Pipeline, zwany inaczej potokiem umożliwia sekwencyjne zastosowanie listy transformatorów do wstępnego przetwarzania danych i w razie potrzeby, zakończenie sekwencji ostatecznym predyktorem do modelowania predykcyjnego (scikit-learn.org, 2024). Narzędzie to pozwala na przetwarzanie w sposób sekwencyjny, uporządkowany w obrębie wielu kroków. Zaletą tego podejścia jest automatyzacja przepływu pracy przekazanie danych wyjściowych z jednego kroku (np. transformacja) do kolejnego (np. klasyfikacja) (rysunek 30). Upraszcza to proces, zwiększa czytelność kodu i zmniejsza ryzyko błędów. Potoki zapewniają spójność danych, a łatwość w ich aplikacji jest dodatkowym atutem przez czytelną dokumentację i łatwość wdrożenia zostały wybrane w tej pracy.

A diagram of a data flow

Description automatically generated

Rysunek 30. Proponowany model systemu przewidywania chorób serca. Źródło: G. N. Ahmad et al.: Efficient Medical Diagnosis of Human Heart Diseases Using Machine Learning Techniques, 2022

W pracy zastosowano także regularyzację, która polega na dodaniu do funkcji kosztu, używanej podczas treningu modelu – kary. Regularyzacja czyni małe modyfikacje algorytmu uczenia się, aby zapewnić lepsze uogólnianie modelu. To pomaga w poprawieniu wydajności modelu na niewidzianych wcześniej danych.

Do automatycznego wyboru hiperparametrów modelu została wybrana funkcja GridSearchCV, gdzie dodatkowo zostały wybrane parametry siły regularyzacji („C”), rodzaj regularyzacji („penalty”) oraz „solver” wspierający regularyzacji, a metryką do oceny zdolności modelu jest „roc\_auc”, która został wybrana na podstawie wiedzy eksperckiej. Wyszukiwanie siatki zapewniane przez GridSearchCV wyczerpująco generuje kandydatów na podstawie siatki wartości parametrów określonych za pomocą parametru param\_grid (scikit-learn.org, 2024). Aby użyć tego podejścia programista na początku definiuje przestrzeń parametrów, które mają być przetestowane w postaci słownika. Kluczem reprezentuje jeden hiperparamter, a wartość to lista, która będzie testowana. Następnie GridSearchCV przeszukuje wskazaną siatkę hiperparametrów z walidacją krzyżową, co umożliwia na dokładniejszą miarę wydajności modelu. Na końcu następuje ocena klasyfikatorów na podstawie wybranej metryki i wybór najlepszego modelu. GridSearch jest bardzo cenionym narzędziem, ponieważ zapewnia systematyczność, pozwala znaleźć najlepsze parametry, automatyzuje proces i zapobiega nadmiernemu dopasowaniu poprzez optymalizację parametrów.

Efektywne budowanie modelu uczenia maszynowego wymaga nie tylko starannego doboru i konfiguracji, ale również rzetelnego oceniania ich wydajności. Określenie odpowiednich wskaźników oceny w celu oceny wydajności modeli AI/ML jest nie tylko konieczne, ale także uzasadnione (Naser & Alavi, 2021). Optymalizacja hiperparametrów jest ściśle powiązana z oceną oraz w sposób bezpośredni wpływa na osiągnięcie wspólnymi siłami optymalnego modelu.

Metryki oceny definiują kryterium, na podstawie którego wybierany jest najlepszy model. Wybór ich zależy od specyfiki problemu, dostępnym zasobów i posiadanych danych. Najczęściej stosowanymi metrykami są Accuracy, Precision, Recall, F1 score i ROC AUC.

Macierz błędów (ang. confusion matrix), łączy w sobie kilka metod oceny i jest narzędziem, który wykorzystuje się w ocenie wydajności modeli klasyfikacyjnych.

* Klasa rzeczywista (ang. actual class): Kategorie, do których faktycznie należą przypadki/obserwacje.
* Przewidywana klasa (ang. predicted class): Kategorie, do których model przewidział przynależność przypadków/obserwacji.

Table

Description automatically generated with medium confidence

Rysunek 31. Macierz błędów z dodatkowymi informacjami o metrykach. Źródło: opracowanie własne

Macierz składa się z czterech kwadrantów (rysunek 31):

* Prawdziwie pozytywne (ang. true positive, w skórcie TP): Przypadki, które są pozytywne i zostały prawidłowo zidentyfikowane jako pozytywne przez model.

Przykład: Pacjent ma chorobę (pozytywny przypadek) i model poprawnie przewiduje, że pacjent jest chory.

* Fałszywie negatywne (ang. false negative, w skrócie FN), znane również jako Błąd Typu II: Przypadki, które są pozytywne, ale zostały błędnie zidentyfikowane jako negatywne przez model.

Przykład: Pacjent ma chorobę, ale model błędnie przewiduje, że pacjent jest zdrowy.

* Fałszywie pozytywne (ang. false positive, w skrócie FP), znane również jako Błąd Typu I: Przypadki, które są negatywne, ale zostały błędnie zidentyfikowane jako pozytywne.

Przykład: Pacjent jest zdrowy (negatywny przypadek), ale model błędnie przewiduje, że pacjent jest chory.

* Prawdziwie negatywne (ang. true negative, w skrócie TN): Przypadki, które są negatywne i zostały prawidłowo zidentyfikowane jako negatywne przez model.

Przykład: Pacjent jest zdrowy i model poprawnie przewiduje, że pacjent nie ma choroby.

Dodatkowo, na grafice przedstawiono kluczowe metryki wydajności modelu:

Czułość (ang. Sensitivity), znana również jako True Positive Rate lub Recall: Odsetek prawidłowo zidentyfikowanych przypadków pozytywnych.

(3.3)

Swoistość (ang. Specificity): Odsetek prawidłowo zidentyfikowanych przypadków negatywnych.

(3.4)

Precyzja (ang. Precision): Odsetek przypadków rzeczywiście pozytywnych spośród wszystkich przypadków zidentyfikowanych jako pozytynwe przez model.

(3.5)

Wartość predykcyjna negatywna (ang. Negative Predictive Value): Odsetek przypadków rzeczywiście negatywnych spośród wszystkich przypadków zidentyfikowanych jako negatywne przez model.

(3.6)

Dokładność (ang. Accuracy): Ogólny odsetek prawidłowo zidentyfikowanych przypadków (pozytywnych i negatywnych).

(3.7)

Macierz błędów oraz powiązane z nią metryki mają duże znaczenie w procesie oceny i przeprowadzeniu porównania modeli klasyfikacyjnych. Miary te pozwalają zrozumieć charakterystykę operacyjną modelu, szczególnie w kontekście fałszywych alarmów (FP) i pominiętych prawdziwych przypadków (FN). Przypadki te mogą mieć wielorakie konsekwencje w zależności od zastosowania modelu, na przykład w systemach bezpieczeństwa komputerowego, może to prowadzić do ignorowania rzeczywistych intruzów z powodu nadmiaru fałszywych alarmów o zagrożeniach. Dlatego sprawdzanie wskaźników, pozwala na dostosowanie modelu do specyfiki dziedziny, w której ma być zaimplementowany.

Warto także zauważyć, że w tym zagadnieniu niebywale istotna jest wiedza ekspercka z dziedziny uczenia maszynowego, jak i również posiadanie obszernej ekspertyzy domenowej, ponieważ kluczowe jest identyfikowanie i zarządzanie kompromisami między różnymi metrykami oceny. Wartości, które są priorytetowe, mogą się różnić w zależności od specyfiki i celów konkretnego projektu. W przypadku identyfikacji osób chorych, poprzez analizę ich wyników badań, pominięcie prawdziwego przypadku będzie nieakceptowalne, więc model ten powinien charakteryzować się wysoką czułością. Natomiast w sytuacji, gdzie alarmowanie jest wysoce kosztowne lub szkodliwe, warto stworzyć model z wysoką precyzją, aby nie budzić niepotrzebnego niepokoju odbiorców tych komunikatów. Dlatego niezbędne jest zrównoważone podejście do wyboru i optymalizacji miar, aby osiągnąć najbardziej odpowiedni model dla zadanego problemu branżowego.

Po wyznaczeniu optymalnych parametrów modele są wykorzystywane do przewidywania na zbiorze walidacyjnym i wykonywane są badania metryk, na których podstawie końcowo zostanie wyłoniony najlepszy model.

Poszukiwania najlepszego modelu to zadanie, które starają się wykonać badacze. Aby tego dokonać naukowcy i praktycy ciągle poszerzają swoją wiedzę na temat dostępnych narzędzi, aby dogłębnie zrozumieć obszar problemu oraz dane, posiadać sprzęt, który będzie miał wystarczające parametry techniczne czy nawet zmieniać podejścia w sposób eksperymentalny. Pomimo tych wszelkich starań, na końcu zawsze jest postawione pytanie, jak dokonać wyboru optymalnego modelu.

W tej pracy zastosowano iteracyjny proces, który integruje techniki oceny modeli z zaawansowanymi metodami selekcji, aby wyłonić model o najbardziej efektywnych i odpornych na nieznane scenariusze parametrach.

Pierwszym etapem jest zbieranie kluczowych metryk wydajności dla każdego z przetestowanych modeli. Metryki te obejmują między innymi precision, recall, f1\_score, roc\_auc, które są zbierane i zapisywane w słowniku „model\_metrics”. Wszystkie te sposoby oceny są niezbędne do sprawdzenia jakości klasyfikacji. W sposób wielopoziomowy dostarczają informacji zarówno o zdolności wykonania poprawnej predykcji, ale również mówią o odporności na błędy wynikające z szumu, niezbilansowania klas czy zniekształceń.

Końcowy wybór modelu jest realizowany przez zaimplementowaną funkcję „select\_best\_model()”, która wykorzystuje metodę ważonej oceny agregowanej na podstawie wcześniej zdefiniowanych wag dla każdej z metryk. Ustalone wagi odzwierciedlają priorytety różnych aspektów wydajności modelu, umożliwiając dostosowanie powstałego procesu selekcji do wymagań specyficznych dla projektu i danych. Precyzja, która jest miarą dokładności pozytywnych predykcji. Waga jej przypisana wskazuje, że jest ona ważna, ale nie dominująca w procesie oceny modelu. Jest to szczególnie istotne w sytuacjach, gdzie koszt fałszywie pozytywnych wyników jest wysoki. Czułość mierzy zdolność do wykrywania rzeczywistych przypadków pozytywnych. Wybrana waga oznacza, że zdolność modelu do wykrywania wszystkich pozytywów jest równie ważna jak jego precyzja. W analizie sentymentu z mediów społecznościowych wiele razy tak samo ważne jest zarówno zminimalizowanie fałszywych pozytywów jak i maksymalizacja zdolności wykrywania rzeczywistych przypadków. Na przykład, kiedy klient napisze negatywną opinię i zostanie ona pomięta, to może to prowadzić do braku reakcji na problem, a następnie wpłynąć na reputację marki. Miara F1 jest harmoniczną średnią precyzji i czułości. Jest to zbalansowana metryka, która jest szczególnie przydatna, gdy potrzebujemy równowagi między precyzją a czułością, a klasy są niezrównoważone. Największa waga przydzielona tej metryce podkreśla jej znaczenie w kontekście badania, sugerując, że zrównoważona zdolność do klasyfikacji jest kluczowym kryterium wyboru modelu. AUC dla krzywej ROC jest miarą zdolności zadanego modelu do różnicowania między dostępnymi klasami przy różnych progach klasyfikacji. Gdy wynik dla AUC jest wysoki, oznacza to, że model dobrze radzi sobie z rozróżnianiem między klasami. Obrana waga dla AUC wskazuje, że ta zdolność do rozróżniania jest ważna, ale nie najważniejsza w kontekście całkowitej oceny modelu. Waga AUC powinna być podwyższona, gdy w analizie głównym celem byłoby analizowanie różnych poziomów sentymentu. Obrane podejście pozwala na agregację metryk w jedną zdolności modelu.

Funkcja ma za zadanie obliczyć znormalizowane wyniki dla każdego modelu, sumując je zgodnie z przypisanymi wagami, co jest tożsame ze stworzeniem obiektywnej i spójnej oceny. Model, który osiągnie największy wynik jest wybieramy jako najbliższy optymalnemu ze wszystkich stworzonych. Po wybraniu go, jest on ładowany za pomocą nazwy z zapisanych lokalnie modeli i używany do dalszych predykcji na zbiorze testowym. W ramach projektu zastosowano dwa podejścia do zapisywania i ładowania modeli, które umożliwiają reutilizację i zarządzenie wytrenowanymi modelami, wygenerowanymi podczas eksperymentów. Dla modeli zbudowanych przy użyciu biblioteki Keras są one zapisywane jako katalogi SavedModel. Metoda ta pozwala na zapisywanie nie tylko architektury modelu, ale również wag i stanu optymalizatora. Jest to przydatne dla modeli uczenia głębokiego, które charakteryzuje często złożona struktura i duże wymagania obliczeniowe podczas treningu. Dla modeli powstałych przy użyciu biblioteki scikit-learn, zastosowano powszechnie praktykowany moduł pickle. Technika ta umożliwia serializację i deserializację obiektów Pythona w tym modeli ML. Po załadowaniu najlepszego modelu, aplikacja jest gotowa do przeprowadzenia predykcji na nowych danych. Wykorzystując nazwę modelu zapisaną w słowniku najlepszego modelu, proces ładowania i predykcji może być w pełni zautomatyzowany, co znacząco zwiększa efektywność zarządzania modelami oraz umożliwia łatwe przeprowadzenie testów na różnych zestawach danych. Automatyzacja ta jest kluczowa w dynamicznych środowiskach, gdzie szybka adaptacja modeli do zmieniających się danych jest często wymagana.

Podsumowując, wyłonienie optymalnego modelu z dostępnych za pomocą przyjętego podejścia pozwala na systematyczną i metodologiczną poprawną ocenę możliwość kandydatów. Obrany sposób zapewnia bez stronniczą, teoretycznie uzasadnioną ocenę, a dzięki temu końcowe przewidywanie klas na zbiorze testowym będzie wykonane efektywnie.

## 3.3 Interpretacja i dyskusja nad wynikami Analizy Sentymentu

W ramach pracy zrealizowano kompleksowe badanie dotyczące analizy sentymentu w danych z mediów społecznościowych. Celem pracy było stworzenie modeli uczenia maszynowego zdolnych do efektywnego rozróżnienia pozytywnych i negatywnych komentarzy. Do osiągnięcia tego celu wykorzystano zaawansowane techniki przetwarzania języka naturalnego oraz uczenia maszynowego co pozwoliło na optymalne modelowanie i interpretacje.

W oparciu o metodę opisaną w poprzednim rozdziale, model konwolucyjnej sieci neuronowej (w skrócie CNN) został wybrany jako najlepszy z kandydatów (tabela 4).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | AUC | precision | recall | F1 score | ROC AUC |
| CNN\_reg\_dropout | 0. 9995 | 0.9955 | 0.9963 | 0.9959 | 0.9951 |
| LOGISTIc\_Reg\_tfidf | 0,9997 | 0,9934 | 0,9909 | 0,9922 | 0,9911 |
| GRU\_dropout | 0,9991 | 0,9903 | 0,9933 | 0,9918 | 0,9902 |
| RNN\_ regularization | 0,9991 | 0,9881 | 0,9940 | 0,9910 | 0,9890 |
| GRU\_gridSearch | 0,9988 | 0,9888 | 0,9903 | 0,9895 | 0,9877 |
| LSTM\_ regularization | 0,9992 | 0,9830 | 0,9955 | 0,9892 | 0,9863 |

Tabela 4. Metryki oceny najlepszego modelu klasyfikującego. Źródło: opracowanie własne

Model CNN\_REG\_DROPOUT został skonstruowany przy użyciu architektury sekwencyjnej z warstwą osadzania (ang. embedding layer) o wymiarze wejściowym równym rozmiarowi słownictwa, wymiarze wyjściowym 50 i długości wejściowej określonej przez maksymalną długość sekwencji. Zastosowano warstwę konwolucyjną z 32 filtrami, jądrem o rozmiarze 3 i funkcją aktywacji "relu". Po warstwie konwolucyjnej zastosowano warstwę porzucania (ang. dropout layer), ale ponieważ najlepszy współczynnik porzucania dla warstwy konwolucyjnej wynosił 0,0, nie zastosowano go. Następnie zastosowano warstwę globalnego łączenia maksymalnego (ang. global max pooling layer) w celu podsumowania konwolucyjnych map funkcji.

W kolejnym kroku dołączono gęstą warstwę (ang. dense layer) z 10 jednostkami, wykorzystującą funkcję aktywacji "relu" i regulowaną normami L1 i L2, obie ustawione na 0,01. Kolejna warstwa dropout została zastosowana po warstwie gęstej, ale ponieważ najlepszy współczynnik wynosił 0,0, tutaj również nie zaaplikowano go. Wreszcie, warstwa wyjściowa składała się z pojedynczego neuronu z sigmoidalną funkcją aktywacji do klasyfikacji binarnej. Model został skompilowany przy użyciu optymalizatora Adam ze współczynnikiem uczenia 0,01 i binarną entropią krzyżową (ang. binary cross-entropy) jako funkcją straty. Hiperparametry zostały zoptymalizowane przy użyciu podejścia grid search w odniesieniu do straty walidacyjnej (ang. validation loss), z maksymalnie trzema próbami (ang. trials).

Konkretny wybór parametrów, w tym 32 filtry w warstwie konwolucyjnej, rozmiar jądra 3, 10 jednostek w gęstej warstwie i wartości regularyzacji 0,01 zarówno dla L1, jak i L2, przyczyniły się do sukcesu modelu, zapewniając równowagę między złożonością modelu a regularyzacją. Brak porzucania, ze wskaźnikami ustawionymi na 0,0, wskazywał, że model nie wymagał dodatkowej regularyzacji poprzez porzucanie, prawdopodobnie ze względu na skuteczność technik regularyzacji L1 i L2. Wybrany współczynnik uczenia 0,01 zapewnił wydajną i stabilną zbieżność podczas treningu. Te starannie dobrane parametry wspólnie zwiększyły zdolność modelu do dobrego uogólniania na danych testowych, co doprowadziło do jego wyjątkowej wydajności w zadaniach klasyfikacji nastrojów (tabela 5).

|  |  |  |
| --- | --- | --- |
| index | comment | predicted\_class |
| 0 | " I don't care what anyone says, I like Hillar... | 0 |
| 1 | have an awesome time at purdue!.. | 1 |
| 2 | Yep, I'm still in London, which is pretty awes... | 1 |
| 3 | Have to say, I hate Paris Hilton's behavior bu... | 0 |
| 4 | i will love the lakers. | 1 |
| ... | ... | ... |
| 33047 | Love Story At Harvard [ awesome drama! | 1 |
| 33048 | well, i had a piece of crap toyota celica but ... | 0 |
| 33049 | But I miss Boston. | 1 |
| 33050 | i love paris hilton.. | 1 |
| 33051 | I was rejected by the stupid San Francisco lit... | 0 |

Tabela 5. Wyniki predykcji najlepszego modelu. Źródło: opracowanie własne

Model CNN jest przystosowany do przetwarzania danych sekwencyjnych i testowych, dlatego mogło to stanowić jeden z powodów jego wygranej. Charakteryzuje się też skłonnością do wydobywania lokalnych i hierarchicznych wzorców. Wykorzystanie tuningu hiperparametrów pozwoliło na dokładne dostosowanie do zadanych danych wejściowych i poprawiło wydajność. Zastosowanie regularyzacji sprawiło, że model jest mało wrażliwy na specyficzne próbki danych, jest to szczególnie ważne w analizie danych z mediów społecznościowych, ponieważ charakteryzując się one różnorodnością kontekstów i stylów pisania. Model ten wykazał się wydajnością na tle innych kandydatów szczególnie w kontekście metryki AUC, co podkreśla jego stosowność w zorientowanych aplikacjach biznesowych analizy sentymentu.

Model spełnia swoją rolę w rozróżnianiu sentymentu, co zostało przedstawione w wizualizacji chmury słów (rysunek 32).

|  |  |
| --- | --- |
| **Chmura słów dla negatywnych komentarzy**  **A word cloud with red text  Description automatically generated** | **Chmura słów dla pozytywnych komentarzy**  **A word cloud with green text  Description automatically generated** |

Rysunek 32. Chmura słów dla oddzielnych kategorii sentymentów. Źródło: opracowanie własne

Chmura słów dla komentarzy negatywnych jest oznaczona paletą czerwieni i można wnioskować, że komentarze przewidziane jako negatywne zawierają często słowa emocjonalnie pejoratywne: „*hate*”, „*bad*”. Kolorami zieleni kreśli się chmura komentarzy pozytywnych, a w niej można dostrzec, że słowa takie jak „*love*”, czy „*awesome*” są częstymi wyrazami w tychże opiniach.

Następnie stworzono zintegrowany obraz porównania sentymentu komentarzy przed i po predykcji modelu (rysunek 33). Przedstawiona grafika, która składa się z dwóch chmur słów reprezentujących najczęściej występujące słowa w komentarzach z zestawu danych treningowych i testowych. Kolory słów odzwierciedlają dominujący sentyment: zielony oznacza pozytywny, czerwony negatywny, a szary mieszany sentyment. W ilustracji został użyty próg = 0.9 który oznacza, że słowo zostanie pokolorowane na zielono (pozytywne) lub czerwono (negatywne) tylko wtedy, gdy 90% lub więcej wszystkich wystąpień tego wyrazu w komentarzach ma odpowiedni sentyment. Jeśli żadne z tych kryteriów nie zostanie spełnione, słowo zostanie pokolorowane na szaro, co wskazuje na mieszany sentyment. Wizualizacja ta pozwala na natychmiastową ocenę charakterystycznych słów. Można z niej wnioskować, że obie chmury słów wykazują spójność w identyfikowaniu dominujących sentymentów dla poszczególnych słów, co sugeruje, że model dobrze nauczył się wzorców z danych treningowych. W kolejnym etapie badania można skoncentrować się na przyczynach częstych komentarzy związanych z konkretnymi osobami czy wydarzeniami. Dla firm i branży filmowej, analiza ta może pomóc w formowaniu bardziej skutecznych strategii komunikacji z widownią.

A close up of words

Description automatically generated A word cloud with different words

Description automatically generated with medium confidence

Rysunek 33. Chmury słów przed i po predykcji sentymentu. Analiza danych treningowych i testowych. Źródło: opracowanie własne

Następnie podjęto się analizy zależności między długością komentarza, a jego sentymentem (rysunek 34). Oś X przedstawia długość komentarza w liczbie znaków, natomiast oś Y reprezentuje sentyment komentarza. Rozkład długości komentarza dla danych testowych oznacza, że większość komentarzy, zarówno pozytywnych jak i negatywnych ma długość poniżej 200 słów. Wykresy pokazują podobne trendy dla danych treningowych jak i testowych.

A graph with numbers and lines

Description automatically generated

A diagram of a graph

Description automatically generated

Rysunek 34. Wykres zależności między długością komentarza, a jego sentymentem. Źródło: opracowanie własne

W dalszej analizie można również przeprowadzanie badań nad wykorzystaniem dodatkowych metadanych (np. liczba polubień, odpowiedzi, kontekst rozmowy) w celu poprawy klasyfikacji sentymentu oraz dostarczenie nowych, cennych informacji.

Rozkład długości komentarza w zestawach danych treningowych i testowych jest podobny (rysunek 35). W obu zestawach danych dominują krótkie komentarze, jednak pomimo ogólnie podobnej charakterystyce, to zestaw testowy (pomarańczowy) ma więcej komentarzy w niższych przedziałach, podczas gdy zestaw treningowych (niebieski) wykazuje więcej dłuższych komentarzy. Rozkład długości komentarzy jest wyraźnie skośny w prawo, co jest typowe dla danych tekstowych z mediów społecznościowych, gdzie zwięzłe wyrażanie opinii jest powszechne.

A graph with orange and white bars

Description automatically generated

Rysunek 35. Rozkład długości komentarzy w zbiorach treningowych i testowych. Źródło: opracowanie własne

Rekomendacją, która może wynikać z wykresu, jest to, że warto dokonać dodatkowej analizy jakości danych, ponieważ krótkie komentarze mogą być bardziej podatne na błędy klasyfikacji ze względu na mniejszą ilość informacji. Rozważenie dodatkowych technik, takich jak analiza kontekstu, może poprawić klasyfikację.

Liczba pozytywnych komentarzy po predykcji dla danych testowych to około 20 000, która jest znacznie większa niż liczba negatywnych komentarzy, których jest około 10 000. Wykres słupkowy (rysunek 36) przedstawia rzeczywistą liczbę komentarzy, co pozwala na łatwe porównanie liczny pozytywnych i negatywnych komentarzy. Wartości bezwzględne pokazują, że pozytywne komentarze są bardziej liczebne niż negatywne.

A graph with a red and green rectangle

Description automatically generated

Rysunek 36. Rozkład liczby pozytywnych i negatywnych komentarzy po predykcji. Źródło: opracowanie własne

Wizualizacja ta jest szczególnie ważna, ponieważ pozwala ocenić, czy istnieją słowa, które mogły być błędnie zaklasyfikowane lub które pojawiają się często w obu kategoriach. Obecność ta może wskazywać na potrzebę dalszego dostosowania procesu lub także co ważniejsze, czy model ma umiejętność wykrywania niuansów i sarkazmu, co może być odzwierciedlone w wysokich metrykach, którymi charakteryzuje się model wybrany jako najlepszy w tym badaniu.

Proces tworzenia i wyboru modelu był złożony. Zostały użyte techniki takie jak tokenizacja, lematyzacja oraz wektoryzacja, co zapewniło to solidną podstawę do skutecznego przetwarzania surowego tekstu na formaty, które były efektywnie używane przez modele ML. Wykorzystano różnorodne algorytmy od regresji logistycznej do skompilowanych sieci neuronowych takich jak CNN i RNN, co umożliwiło przetestowanie wielu podejść. Zastosowano tuning hiperparametrów, co pozwoliło na precyzyjne dostosowanie parametrów modeli. Dokładność ta nie stanowiła jednak negatywnego aspektu, ponieważ regularyzacja zapobiegły nadmiernemu przeuczeniu.

Jednak jako część, którą można rozwinąć w dalszych badaniach zauważa się ulepszenie zastosowanych metod do balansowania klas. Metody takie jak nadpróbkowanie czy podpróbkowanie, mogłyby zostać zastosowane, aby poprawić moc predykcyjną modelu. Dodatkowo, jednym z głównych wyzwać NLP jest interpretacja sarkazmu, co nie zawsze jest możliwe z użyciem standardowych technik. Podczas analizy wyników odnotowano, że w przyszłości warto rozważyć implementację technik dedykowanych temu problemowi, takich jak np. Transfer Learning, lub też zebranie danych do treningu ściśle obejmujących ironię. Pomimo zastosowania GridSearchCV, istnieje sposobność, że nie wszystkie hiperparametry zostały optymalnie skonfigurowane, ponieważ testowany był tylko predefiniowany ich zestaw. Należy rozważyć poszerzenie zakresu przeszukiwania lub zastosowanie innych, bardziej złożonych metod optymalizacji, np. Bayesian optimalisation. Skuteczność CNN chodź obecna, może być mniejsza niż nowszych architektur. Obiecującymi aspektami, które należy rozważyć podczas rozwoju aplikacji są typy sieci, takie jak transformery.

Praca ta demonstruje jak wielopoziomowe i zaawansowane kwestie pracy z technikami NLP oraz modelowaniem danych. Pomimo osiągnięć, nie sposób nie zauważyć jak wiele nowych technologii przybywa do zestawu narzędzi badacza danych. Ciągły rozwój dziedziny sztucznej inteligencji dostarcza wartościowej wiedzy i otwiera wiele dróg do eksploracji. Już teraz jesteśmy w stanie wykrywać nastroje osób publikujących w sieci (co dowodzi ta praca), a było to niemożliwe jeszcze jakiś czas temu. Dalsze badania z nowymi rozwiązaniami będą kluczowe dla przekształcania narzędzi w potężniejsze, zdolne do radzenia sobie z rosnącymi wymaganiami współczesnego świata biznesu. W miarę ewolucji technologii, istotne jest, aby społeczność naukowa, kontynuowali eksploracji tych innowacji w wielu scenariuszach. Ciągłe testowanie, ocena i optymalizacja będzie pozwać na wykorzystanie potencjału.

## 3.5 Wymagania techniczne i sprzętowe

W pracy został wykorzystany język programowania Python. Python wyłonił się w ciągu ostatnich kilku dekad jako narzędzie pierwszej klasy do zadań obliczeniowych w naukach ścisłych, w tym do analizy i wizualizacji dużych zbiorów danych (VanderPlas, 2017). Ze względu na wyraźnie rosnące zasoby tworzonych bibliotek oraz wygodę w interpretowalności, jego wykorzystanie w przetwarzaniu danych, budowaniu modeli uczenia maszynowego, jak również ogólna popularność w dziedzinie data science ciągle wzrasta. Oprócz zalet samego języka, społeczność wokół dostępnych narzędzi i bibliotek sprawia, że Python jest szczególnie atrakcyjny dla zastosowań w dziedzinie nauk o danych, uczenia maszynowego oraz obliczeń naukowych (Raschka, Patterson & Nolet, 2020).

Projekt został zrealizowany z użyciem trzeciej wersji Python’a oraz stosując biblioteki takiej jak Matplotlib, Seaborn, Plotly do wizualizacji; Keras i TensorFlow dla modelowania sieci neuronowych, w tym CNN i RNN; Scikit-learn: dla tradycyjnych algorytmów uczenia maszynowego i przetwarzania danych; NLTK: dla przetwarzania języka naturalnego, w tym tokenizacji, lematyzacji i analizy sentymentu. oraz Pandas, NumPy: dla manipulacji i analizy danych. Uruchomienie aplikacji zmusza do zainstalowania wskazanych modułów jako wymaganie wstępne. Opisane tutaj warunki, są dostępna dla użytkowników systemów operacyjnych: Windows, Linux oraz Macintosh. Kod zapisany jest w formacie Jupyter Notebook i może być uruchamiany za pomocą Anaconda Navigator na komputerze personalnym, ale także w sytuacji, gdy lokalne zasoby są niewystarczające, wykonanie kodu jest możliwe za pośrednictwem rozwiązań chmurowych, takich jak: Google Colab.

Jupyter Notebook to oryginalna aplikacja internetowa do tworzenia i udostępniania dokumentów obliczeniowych. Oferuje proste, uporządkowane doświadczenie skoncentrowane na dokumentach (Project Jupyter, b.d.). Notatniki Jupyter uznawane są za nieocenione narzędzie w analizie danych oraz uczeniu maszynowym. Swoją opinię zawdzięczają możliwości bezpośredniej interakcji z kodem, jak również przez natychmiastowy wglądu w wyniki, co ułatwia eksplorację i iteracyjne udoskonalenia podejścia.

Anaconda jest jedną z wielu otwartych platform, które ułatwiają korzystanie z języków programowania open-source (R, Python) do przetwarzania danych, analizy predykcyjnej oraz obliczeń statystycznych (Kadiyala & Kumar, 2017).

Menedżerem pakietów dla Anacody jest Conda. Conda dba o wszystkie zależności i zapewnia, że pakiety są skonfigurowane wstępnie tak, aby działały z innymi bibliotekami, które zostały poprzednio dodane (Bloice & Holzinger, 2016).

W celu umożliwienia współpracy z innymi osobami w przyszłości, całość aplikacji jest dostępna w centralnym repozytorium GitHub. Platforma ta wykorzystuje git jako podstawowy mechanizm do zarządzania wersjami. Narzędzie git pozwala na śledzenie każdej zmiany w kodzie, co umożliwia przechowywanie historii wersji plików i cofanie się do poprzednich stanów w razie potrzeby.

Kombinacja wymienionych technologii umożliwiła opracowanie opisywanej aplikacji, zapewniając pełne wsparcie dla jej realizacji. Wykorzystanie tych narzędzi pozwoliło na skuteczne zarządzanie kodem źródłowym, śledzenie zmian, napisanie programu oraz wykonanie instrukcji.

# Zakończenie

W ramach niniejszej pracy podjęto wyzwanie realizacji dogłębnej analizy sentymentu wypowiedzi zaczerpniętych z platform mediów społecznościowych. Cel badawczy zdefiniowano jako opracowanie oraz implementację zróżnicowanych modeli uczenia maszynowego, które będą wykazywały zdolność do efektywnego rozróżniania emocjonalnego charakteru wypowiedzi. W kontekście tego projektu skorzystano z rozbudowanej metodologii przetwarzania języka naturalnego oraz wielu technik uczenia maszynowego, co pozwoliło na precyzyjną klasyfikację, a końcowo także interpretację zbiorów danych tekstowych.

Podstawowym etapem zbudowanej aplikacji było przygotowanie danych, które obejmowało czyszczenie danych, tokenizację, eliminację słów nieistotnych z punktu widzenia problematyki oraz wektoryzację. Podjęte kroki stworzyły proces, w efekcie którego surowe dane tekstowe zostały przetransformowane w strukturę umożliwiające działanie algorytmów klasyfikacyjnych. Natomiast w ramach budowy modeli uczenia maszynowego zastosowano różnorodne techniki rozpoczynając od regresji logistycznej, przez drzewa decyzyjne, po rozwiązania oparte o sieci neuronowe. Optymalizacja hiperparametrów każdego modelu odbywa się poprzez zastosowanie automatycznego przeszukiwania siatki parametrów, mając na celu poprawę wydajności oraz automatyzację procesu.

Etap ewaluacji modelu został przeprowadzony z wykorzystaniem metryk cieszących się uznaniem szerokiego grona praktyków: precyzji, czułości, miary F1 czy krzywej ROC AUC. Ocena dokładności modelu pozwoliła na porównanie różnych algorytmów tudzież innych konfiguracji tego samego algorytmu w kontekście zadań klasyfikacyjnych. Użyteczność tego kroku jest bezsporna, ponieważ definiuje zdolność do systematycznego oceniania skuteczności oraz zapewnienie zgodności z określonymi celami oraz wymaganiami. W kontekście NLP istotnym aspektem jest sprawiedliwe i pozbawione uprzedzeń względem grup czy tematów przewidywanie, a ewaluacja może pomóc zidentyfikować i skorygować takie uprzedzenia.

Proces selekcji najoptymalniejszego modelu z puli kandydatów opierał się na analizie zbalansowanych wyników metryk, wykorzystując ich ważone wartości. Model o najlepszych parametrach jest wybierany jak końcowy klasyfikator i jest następnie poddany dodatkowym testom w celu zapewnienia jego stabilności jak i niezawodności dla nowych zestawów danych.

Wyniki niniejszej pracy wskazują jak znaczący postęp w dziedzinie analizy sentymentu został osiągnięty na przełomie ostatnich lat. Stworzony program efektywnie implementuje algorytmy uczenia maszynowego, które po procesie trenowania dokonują klasyfikacji nastrojów wypowiedzi z mediów społecznościowych. Jest to niebywała umiejętność, która przez lata była przypisywana tylko ludziom. Osiągnięcia niniejszej pracy pokazują znaczny postęp w analizie nastrojów, a opracowany model CNN, oznaczony jako CNN\_REG\_DROPOUT, skutecznie klasyfikuje wypowiedzi w mediach społecznościowych i znacznie przyspiesza badania dzięki zautomatyzowanym, połączonym procesom. Model ten osiągnął wyjątkowe wskaźniki wydajności, w tym AUC 0,9995, precyzję 0,9955, recall 0,9963, wynik F1 0,9959 i ROC AUC 0,9951. Model został zbudowany z warstwy osadzania (ang. embedding layer), a następnie warstwy konwolucyjnej z 32 filtrami, jądrem o rozmiarze 3 i funkcją aktywacji "relu". Globalna warstwa maksymalnego łączenia podsumowała funkcje konwolucyjne. Gęsta warstwa (ang. dense layer) z 10 jednostkami została uregulowana za pomocą L1 i L2, obie ustawione na 0,01. Model został skompilowany przy użyciu optymalizatora Adam z szybkością uczenia się 0,01 i binarną funkcją utraty entropii krzyżowej (ang. cross-entropy loss function). Hiperparametry zostały zoptymalizowane za pomocą wyszukiwania siatkowego (ang. grid search). Dodatkowo jako część stworzonej aplikacji zaimplementowane zostały zautomatyzowane, scalone procesy, które przy użyciu potoków i technik selekcji znacząco przyspieszyły badania, umożliwiając szybsze przeszukiwanie siatek parametrów i iteracyjne udoskonalanie modeli. Dzięki automatyzacji proces ten stał się bardziej efektywny i mniej zależny od manualnej interwencji. Wizualizacja wyników, wykorzystująca techniki takie jak chmury słów umożliwiła wgląd i intuicyjne zrozumienie wyników przez interpretację różnych sentymentów. Ten solidny proces podkreśla znaczenie dostrajania hiperparametrów, regularyzacji i technik automatyzacji w zwiększaniu wydajności klasyfikacji nastrojów.

Podsumowując, niniejsze badanie wykazało skuteczność zaawansowanych metod uczenia maszynowego w analizie nastrojów w mediach społecznościowych. Systemy te zapewniają narzędzia do głębokiego zrozumienia sentymentu i opinii publicznej, otwierając perspektywy dla dalszych badań w zakresie przetwarzania języka naturalnego. Należy jednak pamiętać o kilku ograniczeniach obecnej pracy. Wydajność modelu zależy w dużej mierze od jakości oraz reprezentatywności danych treningowych, które mogą nie odzwierciedlać różnorodności języka na innych platformach mediów społecznościowych. Analiza była ograniczona do tekstów w języku angielskim, co może zawężać możliwość zastosowania rozwiązania do mniejszych obszarów geograficznych. Trening złożonych modeli z obszernym dostrajaniem parametrów wymaga znacznych zasobów obliczeniowych. Ze względu na ograniczenia mocy obliczeniowej komputerów osobistych, do obliczeń wykorzystano rozwiązanie chmurowe, podkreślając zależność od zasobów zewnętrznych, które mogą nie być dostępne dla wszystkich. Ogólne modele nastrojów mogą nie sprawdzać się w przypadku tekstów specyficznych dla danej dziedziny, takich jak dokumenty medyczne lub prawne, w których nastroje są wyrażane na odmienne sposoby. Nastroje są często zależne od kontekstu, bez zrozumienia szerszego uwarunkowania klasyfikacje mogą być błędne. Kolejnym wyzwaniem jest obecność fałszywych, zmanipulowanych lub spamowych danych, co może wypaczać wyniki, prowadząc do mylnych wniosków. Chociaż osiągnięto wysoką dokładność, to interpretowalność modelu pozostaje wyzwaniem, wymagającym dalszych prac w celu poprawy wyjaśnialności prognoz. Przyszłe kierunki badań obejmują badanie bardziej zróżnicowanych zbiorów danych, przeprowadzanie wielojęzycznej analizy nastrojów, optymalizację wydajności obliczeniowej i opracowywanie łatwo interpretowalnych modeli uczenia maszynowego.

# Bibliografia

1. *3.2. Tuning the hyper-parameters of an estimator*. (b.d.). Scikit-Learn. Pobrano 11 maj 2024, z <https://scikit-learn/stable/modules/grid_search.html>
2. Abelard, A., & Sibaroni, Y. (2021). Multi-aspect sentiment analysis on netflix application using latent dirichlet allocation and support vector machine methods. *JURNAL INFOTEL*, *13*, 128–133. <https://doi.org/10.20895/infotel.v13i3.670>
3. Agarwal, B., & Mittal, N. (2016). Machine Learning Approach for Sentiment Analysis. W B. Agarwal & N. Mittal (Red.), *Prominent Feature Extraction for Sentiment Analysis* (s. 21–45). Springer International Publishing. <https://doi.org/10.1007/978-3-319-25343-5_3>
4. Aggarwal, C. C. (Red.). (2014). *Data Classification: Algorithms and Applications* (1st edition). Chapman and Hall/CRC.
5. Andzulis, J. „Mick”, Panagopoulos, N. G., & Rapp, A. (2012). A Review of Social Media and Implications for the Sales Process. *The Journal of Personal Selling and Sales Management*, *32*(3), 305–316.
6. Batista, G., & Monard, M.-C. (2002). A Study of K-Nearest Neighbour as an Imputation Method. W *Hybrid Intelligent Systems, ser Front Artificial Intelligence Applications* (T. 30, s. 260).
7. Bhoir, S., Ghorpade, T., & Mane, V. (2017). *Comparative analysis of different word embedding models*. 1–4. <https://doi.org/10.1109/ICAC3.2017.8318770>
8. Bishop, C. (2011). *Pattern Recognition and Machine Learning*. Springer-Verlag New York Inc., Springer Us. <https://www.bookcity.pl/pattern-recognition-and-machine-learning-christopher-m-bishop/pid/A30517254>
9. Bloice, M. D., & Holzinger, A. (2016). A Tutorial on Machine Learning and Data Science Tools with Python. W A. Holzinger (Red.), *Machine Learning for Health Informatics* (T. 9605, s. 435–480). Springer International Publishing. <https://doi.org/10.1007/978-3-319-50478-0_22>
10. Boiy, E., & Moens, M.-F. (2009). A machine learning approach to sentiment analysis in multilingual Web texts. *Information Retrieval*, *12*(5), 526–558. <https://doi.org/10.1007/s10791-008-9070-z>
11. Bollen, J., Mao, H., & Zeng, X.-J. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, *2*(1), 1–8. <https://doi.org/10.1016/j.jocs.2010.12.007>
12. Bonaccorso, G. (2017). *Machine Learning Algorithms*. Packt Publishing Ltd.
13. Carson, Y., & Maria, A. (1997a). Simulation optimization: Methods and applications. *Proceedings of the 29th Conference on Winter Simulation  - WSC ’97*, 118–126. <https://doi.org/10.1145/268437.268460>
14. Carson, Y., & Maria, A. (1997b). Simulation optimization: Methods and applications. *Proceedings of the 29th conference on Winter simulation*, 118–126. <https://doi.org/10.1145/268437.268460>
15. Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches* (arXiv:1409.1259). arXiv. <http://arxiv.org/abs/1409.1259>
16. Chopra, A., Prashar, A., & Sain, C. (2013). *Natural Language Processing*. *1*(4).
17. Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling* (arXiv:1412.3555). arXiv. <http://arxiv.org/abs/1412.3555>
18. Claight Corp. (b.d.). *United States Pizza Market Report and Forecast 2024-2032*. Pobrano 5 czerwiec 2024, z <https://www.expertmarketresearch.com/reports/united-states-pizza-market>
19. Crowston, K., Allen, E. E., & Heckman, R. (2012). Using natural language processing technology for qualitative data analysis. *International Journal of Social Research Methodology*, *15*(6), 523–543. <https://doi.org/10.1080/13645579.2011.625764>
20. Cunningham, P., & Delany, S. J. (2022). k-Nearest Neighbour Classifiers—A Tutorial. *ACM Computing Surveys*, *54*(6), 1–25. <https://doi.org/10.1145/3459665>
21. Dang, N. C., Moreno-García, M. N., & De la Prieta, F. (2020). Sentiment Analysis Based on Deep Learning: A Comparative Study. *Electronics*, *9*(3), Article 3. <https://doi.org/10.3390/electronics9030483>
22. de Ville, B. (2013). Decision trees. *WIREs Computational Statistics*, *5*(6), 448–455. <https://doi.org/10.1002/wics.1278>
23. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407. [https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9](https://doi.org/10.1002/(SICI)1097-4571(199009)41:6%3c391::AID-ASI1%3e3.0.CO;2-9)
24. Devika, M. D., Sunitha, C., & Ganesh, A. (2016). Sentiment Analysis: A Comparative Study on Different Approaches. *Procedia Computer Science*, *87*, 44–49. <https://doi.org/10.1016/j.procs.2016.05.124>
25. Dongare, A. D., Kharde, R. R., & Kachare, A. D. (2012). *Introduction to Artificial Neural Network*. *2*(1).
26. ED CURRINGTON. (2024, styczeń 12). *15 Pizza Marketing Ideas For Every Pizzeria Business [2024 update]*. <https://thedigitalrestaurant.com/pizza-marketing-ideas/>
27. Eisenstein, J. (2019). *Introduction to Natural Language Processing*. The MIT Oress. <https://mitpress.mit.edu/9780262042840/introduction-to-natural-language-processing/>
28. Enholm, I. M., Papagiannidis, E., Mikalef, P., & Krogstie, J. (2022). Artificial Intelligence and Business Value: A Literature Review. *Information Systems Frontiers*, *24*(5), 1709–1734. <https://doi.org/10.1007/s10796-021-10186-w>
29. Erickson, B. J., & Kitamura, F. (2021). Magician’s Corner: 9. Performance Metrics for Machine Learning                    Models. *Radiology: Artificial Intelligence*, *3*(3), e200126. <https://doi.org/10.1148/ryai.2021200126>
30. Erwig, M., & Gopinath, R. (2012). *Explanations for Regular Expressions*. *7212*, 394–408. <https://doi.org/10.1007/978-3-642-28872-2_27>
31. Fanni, S. C., Febi, M., Aghakhanyan, G., & Neri, E. (2023). Natural Language Processing. W M. E. Klontzas, S. C. Fanni, & E. Neri (Red.), *Introduction to Artificial Intelligence* (s. 87–99). Springer International Publishing. <https://doi.org/10.1007/978-3-031-25928-9_5>
32. Fawcett, T., & Provost, F. (2013). *Data Science for Business*.
33. Garcia, E. K., Feldman, S., Gupta, M. R., & Srivastava, S. (2010). Completely Lazy Learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(9), 1274–1285. <https://doi.org/10.1109/TKDE.2009.159>
34. Gentleman, R., & Carey, V. J. (2008). Unsupervised Machine Learning. W F. Hahne, W. Huber, R. Gentleman, & S. Falcon (Red.), *Bioconductor Case Studies* (s. 137–157). Springer. <https://doi.org/10.1007/978-0-387-77240-0_10>
35. Geron Aurelien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. Concepts, Tools, and Techniques to Build Intelligent Systems. 2nd Edition*. O’Reilly Media. <https://helion.pl/ksiazki/hands-on-machine-learning-with-scikit-learn-keras-and-tensorflow-concepts-tools-and-techniques-auraslien-gasron,e_17oi.htm>
36. Goldberg, Y. (2015). *A Primer on Neural Network Models for Natural Language Processing* (arXiv:1510.00726). arXiv. <https://doi.org/10.48550/arXiv.1510.00726>
37. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
38. Greenacre, M., Groenen, P., Hastie, T., Iodice D’Enza, A., Markos, A., & Tuzhilina, E. (2022). Principal component analysis. *Nature Reviews Methods Primers*, *2*, 100. <https://doi.org/10.1038/s43586-022-00184-w>
39. Grimes, T. (2013, lipiec 24). What the Share a Coke campaign can teach other brands. *The Guardian*. <https://www.theguardian.com/media-network/media-network-blog/2013/jul/24/share-coke-teach-brands>
40. Hansen, P. C., Sekii, T., & Shibahashi, H. (1992). The Modified Truncated SVD Method for Regularization in General Form. *SIAM Journal on Scientific and Statistical Computing*, *13*(5), 1142–1150. <https://doi.org/10.1137/0913066>
41. Hawkins, D. M. (2004). The Problem of Overfitting. *Journal of Chemical Information and Computer Sciences*, *44*(1), 1–12. <https://doi.org/10.1021/ci0342472>
42. He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284. <https://doi.org/10.1109/TKDE.2008.239>
43. He, H., & Ma, Y. (2013). Imbalanced Learning: Foundations, Algorithms, and Applications. *Imbalanced Learning: Foundations, Algorithms, and Applications*. <https://doi.org/10.1002/9781118646106>
44. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
45. *How Tech Giants Harness the Power of Customer Sentiment Analytics*. (2024, luty 20). <https://www.wonderflow.ai/blog/How-Tech-Giants-Harness-the-Power-of-Customer-Sentiment-Analytics/>
46. Hutto, C., & Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *Proceedings of the International AAAI Conference on Web and Social Media*, *8*(1), Article 1. <https://doi.org/10.1609/icwsm.v8i1.14550>
47. Hutto, C. J. (b.d.). *vaderSentiment: VADER Sentiment Analysis. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media, and works well on texts from other domains.* (3.3.2) [Software]. Pobrano 10 maj 2024, z <https://github.com/cjhutto/vaderSentiment>
48. Irie, K., Tüske, Z., Alkhouli, T., Schlüter, R., & Ney, H. (2016). *LSTM, GRU, Highway and a Bit of Attention: An Empirical Overview for Language Modeling in Speech Recognition* (s. 3523). <https://doi.org/10.21437/Interspeech.2016-491>
49. I.T. Jolliffe. (2002). *Principal Component Analysis*. <https://link.springer.com/book/10.1007/b98835>
50. Jain, S. jain. (2018, kwiecień 19). An Overview of Regularization in Deep Learning (with Python code). *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/>
51. James, G., Daniela Witten, Hastie, T., & Robert Tibshirani. (2013). *An introduction to statistical learning with applications in R* (T. 6). Springer Science and Business Media. <https://www.tandfonline.com/doi/full/10.1080/24754269.2021.1980261>
52. Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, *31*(3), 685–695. <https://doi.org/10.1007/s12525-021-00475-2>
53. Jivani, A. G. (2011a). *A Comparative Study of Stemming Algorithms*. *2*.
54. Jivani, A. G. (2011b). *A Comparative Study of Stemming Algorithms*. *2*.
55. Jurafsky, D., & James, H. Martin. (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall.
56. Kadiyala, A., & Kumar, A. (2017). Applications of Python to evaluate environmental data science problems. *Environmental Progress & Sustainable Energy*, *36*(6), 1580–1586. <https://doi.org/10.1002/ep.12786>
57. Kaggle. (2024). W *Wikipedia*. <https://en.wikipedia.org/w/index.php?title=Kaggle&oldid=1219952976>
58. Kang, Y., Cai, Z., Tan, C.-W., Huang, Q., & Liu, H. (2020). Natural language processing (NLP) in management research: A literature review. *Journal of Management Analytics*, *7*(2), 139–172. <https://doi.org/10.1080/23270012.2020.1756939>
59. Kogan, S., Levin, D., Routledge, B., Sagi, J., & Smith, N. (2009). *Predicting Risk from Financial Reports with Regression.* 272–280. <https://doi.org/10.3115/1620754.1620794>
60. Kotsiantis, S. B. (2013). Decision trees: A recent overview. *Artificial Intelligence Review*, *39*(4), 261–283. <https://doi.org/10.1007/s10462-011-9272-4>
61. Kukačka, J., Golkov, V., & Cremers, D. (2017). *Regularization for Deep Learning: A Taxonomy* (arXiv:1710.10686). arXiv. <http://arxiv.org/abs/1710.10686>
62. Kurita, T. (2021). *Principal Component Analysis (PCA)* (s. 1013–1016). <https://doi.org/10.1007/978-3-030-63416-2_649>
63. LaValley, M. P. (2008). Logistic Regression. *Circulation*, *117*(18), 2395–2399. <https://doi.org/10.1161/CIRCULATIONAHA.106.682658>
64. Le, Q., & Mikolov, T. (2014). Distributed Representations of Sentences and Documents. *31st International Conference on Machine Learning, ICML 2014*, *4*.
65. Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). *A Critical Review of Recurrent Neural Networks for Sequence Learning* (arXiv:1506.00019). arXiv. <http://arxiv.org/abs/1506.00019>
66. Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers. <https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf>
67. Lopez, M. M., & Kalita, J. (2017). *Deep Learning applied to NLP* (arXiv:1703.03091). arXiv. <http://arxiv.org/abs/1703.03091>
68. Maithani, M. (2021, styczeń 5). *A Comprehensive Guide To 15 Most Important NLP Datasets*. Analytics India Magazine. <https://analyticsindiamag.com/15-most-important-nlp-datasets/>
69. Manning, C., Raghavan, P., & Schuetze, H. (2009). *Introduction to Information Retrieval*.
70. Mikolov, T., Corrado, G. s, Chen, K., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. 1–12.
71. Mikolov, T., Joulin, A., Chopra, S., Mathieu, M., & Ranzato, M. (2015). *Learning Longer Memory in Recurrent Neural Networks* (arXiv:1412.7753). arXiv. <https://doi.org/10.48550/arXiv.1412.7753>
72. Mobayyen, F., & De Almeida, R. (2005). The influence of semantic and morphological complexity of verbs on sentence recall: Implications for the nature of conceptual representation and category-specific deficits. *Brain and cognition*, *57*, 168–171. <https://doi.org/10.1016/j.bandc.2004.08.039>
73. Muhammad, S. H., Abdulmumin, I., Ayele, A. A., Ousidhoum, N., Adelani, D. I., Yimam, S. M., Ahmad, I. S., Beloucif, M., Mohammad, S. M., Ruder, S., Hourrane, O., Brazdil, P., Ali, F. D. M. A., David, D., Osei, S., Bello, B. S., Ibrahim, F., Gwadabe, T., Rutunda, S., … Arthur, S. (2023). *AfriSenti: A Twitter Sentiment Analysis Benchmark for African Languages* (arXiv:2302.08956; Wersja 5). arXiv. <http://arxiv.org/abs/2302.08956>
74. Naser, M. Z., & Alavi, A. H. (2023). Error Metrics and Performance Fitness Indicators for Artificial Intelligence and Machine Learning in Engineering and Sciences. *Architecture, Structures and Construction*, *3*(4), 499–517. <https://doi.org/10.1007/s44150-021-00015-8>
75. Nasteski, V. (2017). An overview of the supervised machine learning methods. *HORIZONS.B*, *4*, 51–62. <https://doi.org/10.20544/HORIZONS.B.04.1.17.P05>
76. Netflix. (b.d.). *Machine Learning Scientist (L4/5)—Content & Media ML Foundations*. Pobrano 4 czerwiec 2024, z <https://jobs.netflix.com/jobs/308370003>
77. Ng, A. (2018). *Machine Learning Yearning*. deeplearning.ai.
78. Ni, R., & Cao, H. (2020). Sentiment Analysis based on GloVe and LSTM-GRU. *2020 39th Chinese Control Conference (CCC)*, 7492–7497. <https://doi.org/10.23919/CCC50068.2020.9188578>
79. *NLTK :: Natural Language Toolkit*. (b.d.). Pobrano 10 maj 2024, z <https://www.nltk.org/>
80. O’Shea, K., & Nash, R. (2015). *An Introduction to Convolutional Neural Networks* (arXiv:1511.08458). arXiv. <http://arxiv.org/abs/1511.08458>
81. P. Adby, M. A. H. DEMPSTER. (2013). *Introduction to Optimization Methods*. Springer Science & Business Media.
82. Pak, A., & Paroubek, P. (2010). *Twitter as a Corpus for Sentiment Analysis and Opinion Mining*. *10*, 1320–1321. <https://www.researchgate.net/publication/220746311_Twitter_as_a_Corpus_for_Sentiment_Analysis_and_Opinion_Mining>
83. Pang, B., & Lee, L. (2008). *Opinion mining and sentiment analysis*. *2*(1–2), 20.
84. Patodkar, V. N., & I.R, S. (2016). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *IJARCCE*, *5*(12), 321–322. <https://doi.org/10.17148/IJARCCE.2016.51274>
85. Pennington, J., Socher, R., & Manning, C. (2014). *Glove: Global Vectors for Word Representation*. *14*, 1532–1543. <https://doi.org/10.3115/v1/D14-1162>
86. Per Christian Hansen. (1987). *The truncatedSVD as a method for regularization*. Springer. <https://link.springer.com/article/10.1007/bf01937276>
87. Pinker, S. (2007). *The Language Instinct: How the Mind Creates Language*. Harper Perennial. <https://www.amazon.pl/Language-Instinct-How-Mind-Creates/dp/0061336467>
88. Poggio, T., Torre, V., & Koch, C. (1985). Computational vision and regularization theory. *Nature*, *317*(6035), 314–319. <https://doi.org/10.1038/317314a0>
89. Project Jupyter’s. (2024). *Dokumetacja oprogramowania Jupyter Notebook* [Software]. <https://jupyter.org>
90. Ranstam, J., & Cook, J. A. (2018). LASSO regression. *British Journal of Surgery*, *105*(10), 1348. <https://doi.org/10.1002/bjs.10895>
91. Raschka, S., Patterson, J., & Nolet, C. (2020). Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence. *Information*, *11*(4), 193. <https://doi.org/10.3390/info11040193>
92. Reis, J. C. S., Correia, A., Murai, F., Veloso, A., & Benevenuto, F. (2019). Supervised Learning for Fake News Detection. *IEEE Intelligent Systems*, *34*(2), 76–81. <https://doi.org/10.1109/MIS.2019.2899143>
93. S Visa, B Ramsay, AL Ralescu, E Van Der Knaap. (2011). Confusion matrix-based feature selection. *Maics*. [https://www.researchgate.net/profile/Jennifer-Seitzer-2/publication/220833258\_Using\_a\_Genetic\_Algorithm\_to\_Evolve\_a\_D\_Search\_Heuristic/links/545a2bea0cf2bccc49132577/Using-a-Genetic-Algorithm-to-Evolve-a-D-Search-Heuristic.pdf#page=126](https://www.researchgate.net/profile/Jennifer-Seitzer-2/publication/220833258_Using_a_Genetic_Algorithm_to_Evolve_a_D_Search_Heuristic/links/545a2bea0cf2bccc49132577/Using-a-Genetic-Algorithm-to-Evolve-a-D-Search-Heuristic.pdf" \l "page=126)
94. Salehin, I., & Kang, D.-K. (2023). A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain. *Electronics*, *12*(14), Article 14. <https://doi.org/10.3390/electronics12143106>
95. Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2018). *Recent Advances in Recurrent Neural Networks* (arXiv:1801.01078). arXiv. <http://arxiv.org/abs/1801.01078>
96. Satrajit Basu. (b.d.). *Sentiment Analysis: How Amazon Aurora Machine Learning and Comprehend Can Revolutionize Customer Review Analysis - DZone*. Dzone.Com. Pobrano 3 czerwiec 2024, z <https://dzone.com/articles/sentiment-analysis-how-amazon-aurora-machine-learn>
97. Savova, G. K., Masanz, J. J., Ogren, P. V., Zheng, J., Sohn, S., Kipper-Schuler, K. C., & Chute, C. G. (2010). Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): Architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, *17*(5), 507–513. <https://doi.org/10.1136/jamia.2009.001560>
98. *Share a Coke campaign post-analysis | Marketing Mag*. (2012, czerwiec 22). <https://www.marketingmag.com.au/news/share-a-coke-campaign-post-analysis/>
99. Sharma, A., & Dey, S. (2012). A comparative study of feature selection and machine learning techniques for sentiment analysis. *Proceedings of the 2012 ACM Research in Applied Computation Symposium*, 1–7. <https://doi.org/10.1145/2401603.2401605>
100. Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, 132306. <https://doi.org/10.1016/j.physd.2019.132306>
101. Shiri, F. M., Perumal, T., Mustapha, N., & Mohamed, R. (b.d.). *A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU*.
102. *Sklearn.decomposition.TruncatedSVD*. (b.d.). Scikit-Learn. Pobrano 10 maj 2024, z <https://scikit-learn/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html>
103. *Sklearn.pipeline.Pipeline*. (b.d.). Scikit-Learn. Pobrano 11 maj 2024, z <https://scikit-learn/stable/modules/generated/sklearn.pipeline.Pipeline.html>
104. Smagulova, K., & James, A. P. (2019). A survey on LSTM memristive neural network architectures and applications. *The European Physical Journal Special Topics*, *228*(10), 2313–2324. <https://doi.org/10.1140/epjst/e2019-900046-x>
105. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (b.d.). *Dropout: A Simple Way to Prevent Neural Networks from Overﬁtting*.
106. Sun, S., Cao, Z., Zhu, H., & Zhao, J. (2020). A Survey of Optimization Methods From a Machine Learning Perspective. *IEEE Transactions on Cybernetics*, *50*(8), 3668–3681. <https://doi.org/10.1109/TCYB.2019.2950779>
107. Sutskever, I., Martens, J., & Hinton, G. (b.d.). *Generating Text with Recurrent Neural Networks*.
108. Team TBH. (2023, czerwiec 9). *Branding Case Study: Success of Share A Coke Campaign*. The Brand Hopper. <https://thebrandhopper.com/2023/06/09/branding-case-study-success-of-share-a-coke-campaign/>
109. Thakor, P., & Sasi, S. (2015). Ontology-based Sentiment Analysis Process for Social Media Content. *Procedia Computer Science*, *53*, 199–207. <https://doi.org/10.1016/j.procs.2015.07.295>
110. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., & Kappas, A. (2010). Sentiment Strength Detection in Short Informal Text. *Journal of the American Society for Information Science and Technology*, *61*, 2544–2558. <https://doi.org/10.1002/asi.21416>
111. Todd Escalona. (2018, styczeń 26). *Detect sentiment from customer reviews using Amazon Comprehend | AWS Machine Learning Blog*. <https://aws.amazon.com/blogs/machine-learning/detect-sentiment-from-customer-reviews-using-amazon-comprehend/>
112. *UMICH SI650—Sentiment Classification*. (b.d.). Pobrano 9 maj 2024, z <https://kaggle.com/competitions/si650winter11>
113. Usama, M., Qadir, J., Raza, A., Arif, H., Yau, K. A., Elkhatib, Y., Hussain, A., & Al-Fuqaha, A. (2019). Unsupervised Machine Learning for Networking: Techniques, Applications and Research Challenges. *IEEE Access*, *7*, 65579–65615. <https://doi.org/10.1109/ACCESS.2019.2916648>
114. Valdivia, A., Luzon, M., & Herrera, F. (2017). *Sentiment Analysis on TripAdvisor: Are There Inconsistencies in User Reviews?* 15–25. <https://doi.org/10.1007/978-3-319-59650-1_2>
115. VanderPlas, J. (2016). *Python Data Science Handbook: Essential Tools for Working with Data*. O’Reilly Media, Inc.
116. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. ukasz, & Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems*, *30*. <https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>
117. Ward, R., Wu, X., & Bottou, L. (2020). AdaGrad stepsizes: Sharp convergence over nonconvex landscapes. *Journal of Machine Learning Research*, *21*(219), 1–30.
118. Webster, J. J., & Kit, C. (1992). Tokenization as the Initial Phase in NLP. *COLING 1992 Volume 4: The 14th International Conference on Computational Linguistics*. COLING 1992. <https://aclanthology.org/C92-4173>
119. Wei, C., Kakade, S., & Ma, T. (2020). The Implicit and Explicit Regularization Effects of Dropout. *Proceedings of the 37th International Conference on Machine Learning*, 10181–10192. <https://proceedings.mlr.press/v119/wei20d.html>
120. Wu, J., Chen, X.-Y., Zhang, H., Xiong, L.-D., Lei, H., & Deng, S.-H. (2019). Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimizationb. *Journal of Electronic Science and Technology*, *17*(1), 26–40. <https://doi.org/10.11989/JEST.1674-862X.80904120>
121. Wu, R. M. X., & Mircea, M. (2021). *E-Business: Higher Education and Intelligence Applications*. BoD – Books on Demand.
122. Xiaogang, W. (2015). *Machine Learning Basics*. <http://whdeng.cn/Teaching/PPT_01_Machine%20learning%20Basics.pdf>
123. Yang, J., & Long, Q. (2024). A modification of adaptive moment estimation (adam) for machine learning. *Journal of Industrial and Management Optimization*, 0–0. <https://doi.org/10.3934/jimo.2024014>
124. Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*, *31*(7), 1235–1270. <https://doi.org/10.1162/neco_a_01199>
125. Zegarra, F. C., Vargas-Machuca, J., & Coronado, A. M. (2023). A Comparative Study of CNN, LSTM, BiLSTM, and GRU Architectures for Tool Wear Prediction in Milling Processes. *Journal of Machine Engineering*. <https://doi.org/10.36897/jme/174019>
126. Zhang, L., Wang, S., & Liu, B. (2018a). Deep learning for sentiment analysis: A survey. *WIREs Data Mining and Knowledge Discovery*, *8*(4), e1253. <https://doi.org/10.1002/widm.1253>
127. Zhang, L., Wang, S., & Liu, B. (2018b). Deep learning for sentiment analysis: A survey. *WIREs Data Mining and Knowledge Discovery*, *8*(4), e1253. <https://doi.org/10.1002/widm.1253>
128. Zhang, Z. (2015). *The Singular Value Decomposition, Applications and Beyond* (arXiv:1510.08532). arXiv. <https://doi.org/10.48550/arXiv.1510.08532>

# Spis rysunków

Rysunek 1. Przykłady klasyfikacji sentymentu w opinii o filmach. Źródło: opracowanie własne

Rysunek 2. Schemat przepływu procesu obsługi klienta w zakresie zgłaszania i rozwiązywania usterek w oprogramowaniu. Źródło: opracowanie własne

Rysunek 3. Porównanie głębokiego uczenia z uczeniem maszynowym. Źródło: opracowanie własne

Rysunek 4. Przykładowy proces automatycznej analizy sentymentu dla treści pochodzących z mediów społecznościowych. Źródło: opracowanie własne

Rysunek 5. Przykłady użycia funkcji CountVectorizer dla różnych dokumentów. Źródło: opracowanie własne

Rysunek 6. Rozkład publikacji dotyczących NLP w prestiżowych periodykach zarządczych. Źródło: Kang, Y., Cai, Z., Tan, C.-W., Huang, Q., & Liu, H. (2020). Natural language processing (NLP) in management research: A literature review. Journal of Management Analytics, 1–34.

Rysunek 7. Przykład procesu tokenizacji. Źródło: opracowanie własne

Rysunek 8. Przykłady stemmatyzacji dla różnych odmian słowa "nauka". Źródło: opracowanie własne

Rysunek 9. Przykłady lematyzacji dla różnych odmian słowa "nauka". Źródło: opracowanie własne

Rysunek 10. Klasyfikacja metod i technik analizy sentymentu. Źródło: Dang, N. C., Moreno-García, M. N., & De la Prieta, F. (2020). Sentiment Analysis Based on Deep Learning: A Comparative Study. Electronics, 9(3), 483.

Rysunek 11. Wykres funkcji sigmoidalnej. Źródło: Kleinbaum, D. G., & Klein, M. (2010). Logistic Regression: A Self-Learning Text (3rd ed.). Springer Science Business Media. ISBN: 978-1-4419-1741-6. DOI 10.1007/978-1-4419-1742-3.

Rysunek 12. Klasyfikacja metodą 3-najbliższych sąsiadów w dwuwymiarowej przestrzeni cech (Monthly\_Sal i Amount). Źródło: Cunningham, P., & Delany, S. J. (2020). k-Nearest Neighbour Classifiers 2nd Edition (with Python examples). arXiv:2004.04523v2 [cs.LG].

Rysunek 13. Drzewo decyzyjne. Źródło: Rozwadowska, A. (2022). Wspomaganie budowania modeli klasyfikacyjnych w przypadku niezbilansowanych klas w zbiorze danych.

Rysunek 14. Model architektury transformera. Źródło: Vaswani, A., Jones, L., Shazeer, N., Parmar, N., Uszkoreit, J., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need.

Rysunek 15. Różnice między dwoma podejściami do klasyfikacji polarności sentymentu: uczenie maszynowe (góra) i głębokie uczenie (dół). Część mowy (ang. Parts-Of-Speech, w skrócie: POS); Rozpoznawanie nazwanych encji (NER); Częstość terminów - odwrotna częstość dokumentów (ang. Term Frequency-Inverse Document Frequency, w skrócie: TF-IDF). Źródło: Dang, N. C., Moreno-García, M. N., & De la Prieta, F. (2020). Sentiment Analysis Based on Deep Learning: A Comparative Study. Electronics, 9(3), 483.

Rysunek 16. Rekurencyjna sieć neuronowa to sieć neuronowa o bardzo głębokim sprzężeniu zwrotnym, która ma warstwę dla każdego kroku czasowego. Jego wagi rozkładają się w czasie. Źródło: Sutskever, I. (2013). Training Recurrent Neural Networks. Doktorat, Graduate Department of Computer Science, University of Toronto.

Rysunek 17. Przykład sieci z ośmioma jednostkami wejściowymi, czterema jednostkami wyjściowymi i dwoma blokami komórek pamięci o rozmiarze 2. *in1* oznacza bramkę wejściową, *out1* oznacza bramkę wyjściową, a *cell1/block1* oznacza pierwszą komórkę pamięci bloku 1. Źródło: Hochreiter, S., & Schmidhuber, J. (1

997). Long Short-Term Memory. Neural Computation, 9(8), 1745.

Rysunek 18. Oryginalna architektura LSTM. Źródło: Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. Neural Computation, 1–36.

Rysunek 19. Architektura LSTM z bramką zapominania. Źródło: Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. Neural Computation, 1–36.

Rysunek 20. Architektura LSTM z GRU. Źródło: Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. Neural Computation, 1–36.

Rysunek 21. Przykładem zastosowania metody Bag of Words Źródło: opracowanie własne

Rysunek 22. Zastosowanie metody dropout w konwencjonalnej sieci neuronowej. Źródło: opracowanie własne

Rysunek 23. Porównanie małego, optymalnego i dużego współczynnika uczenia. Źródło: Jordan (2018)

Rysunek 24. Rozkład sentymentów w danych treningowych. Źródło: opracowanie własne

Rysunek 25. Rozkład gęstości długości komentarzy dla różnych sentymentów. Źródło: opracowanie własne

Rysunek 26. Chmura słów dla terminów występujących w pozytywnych komentarza zestawu danych treningowych. Źródło: opracowanie własne

Rysunek 27. Częstotliwość występowania 10 najczęściej używanych słów w pozytywnych i negatywnych komentarzach. Źródło: opracowanie własne

Rysunek 28. Wybór optymalnego modelu za pomocą podziału danych na trzy zbiory. Źródło: opracowanie własne

Rysunek 29. Wizualizacja dominujących terminów w opiniach. Źródło: opracowanie własne

Rysunek 30. Proponowany model systemu przewidywania chorób serca. Źródło: G. N. Ahmad et al.: Efficient Medical Diagnosis of Human Heart Diseases Using Machine Learning Techniques, 2022

Rysunek 31. Macierz błędów z dodatkowymi informacjami o metrykach. Źródło: opracowanie własne

Rysunek 32. Chmura słów dla oddzielnych kategorii sentymentów. Źródło: opracowanie własne

Rysunek 33. Chmury słów przed i po predykcji sentymentu. Analiza danych treningowych i testowych. Źródło: opracowanie własne

Rysunek 34. Wykres zależności między długością komentarza, a jego sentymentem. Źródło: opracowanie własne

Rysunek 35. Rozkład długości komentarzy w zbiorach treningowych i testowych. Źródło: opracowanie własne

Rysunek 36. Rozkład liczby pozytywnych i negatywnych komentarzy po predykcji. Źródło: opracowanie własne

# Spis tabel

Tabela 1. Złożoność oraz niejednoznaczność sentymentu w przekazach tekstowych. Źródło: opracowanie własne

Tabela 2. Macierz cech po transformacji Term Frequency. Źródło: opracowanie własne

Tabela 3. Macierz cech po transformacji Term Frequency-Inverse Document Frequency. Źródło: opracowanie własne

Tabela 4. Metryki oceny najlepszego modelu klasyfikującego. Źródło: opracowanie własne

Tabela 5. Wyniki predykcji najlepszego modelu. Źródło: opracowanie własne

# Załączniki

Załącznik 1. Program aplikacji napisany w języku Python, załączony w formacie Jupyter Notebook, zatytułowany „Aleksandra\_Gomolka.ipynb”

# Streszczenie

Praca dotyczy analizy sentymentu w mediach społecznościowych z wykorzystaniem zaawansowanych technik przetwarzania języka naturalnego i algorytmów uczenia maszynowego. Cel badania został określony jako opracowanie efektywnych procesów zdolnych do identyfikacji oraz klasyfikacji emocji wyrażonych w tekstach pochodzących z platform społecznościowych. Całość aplikacji zostało zaimplementowanej w oparciu o język programowania Python.

W ramach niniejszej pracy przeprowadzono kompleksowy proces obejmujący przygotowanie danych, modelowanie, optymalizację hiperparametrów oraz ewaluację powstałych modeli klasyfikacyjnych. Stworzona aplikacja zawiera różnorodne techniki, w tym tytułem przykładu: konwolucyjne sieci neuronowe, regularyzację, klasyczne algorytmy uczenia maszynowego, a także metody oceny modeli, takie jak precyzja, czułość, pole pod krzywą ROC. Szczególną uwagę poświęcono doborowi parametrów oraz metodom redukcji wymiarowości, aby zwiększyć skuteczność proponowanych rozwiązań.

Głównym osiągnięciem było skonstruowanie i wyselekcjonowanie najbardziej optymalnego modelu w sposób automatyczny i dokonania klasyfikacji, bez potrzeby ingerencji manualnej badacza. Wybrany model wyróżnia się zdolnością do efektywnego przetwarzania i przewidywania kategorii z danych, co potwierdzono wysokimi wartościami metryk ewaluacyjnych. Ostateczna analiza wyników obejmuje również wizualizację w postaci m.in. chmury słów, ilustracji dominujących wyrazów i fraz zarówno w komentarzach o sentymencie pozytywnym i negatywnym.

Praca kończy się dyskusją na temat osiągnięć, wyzwań tudzież możliwości dalszych badań w dziedzinie analizy sentymentu. Podkreśla znaczenie ciągłego rozwoju narzędzi dostępnych w obszarze uczenia maszynowego, analizy języka naturalnego, czy też wstępnego przetwarzania danych, w kontekście dynamicznie zmieniającego się zapotrzebowania.