Spis treści

[Wstęp 3](#_Toc165960927)

[1 Analiza sentymentu 3](#_Toc165960928)

[1.1 Wprowadzenie do Analizy sentymentu 3](#_Toc165960929)

[1.2 Podstawy Przetwarzania Języka Naturalnego (NLP) 10](#_Toc165960930)

[1.3 Przegląd narzędzi i technologii w analizie sentymentu 15](#_Toc165960931)

[1.4 Potencjalne Wyzwania i Ograniczenia 17](#_Toc165960932)

[2 Uczenie maszynowe w analizie sentymentu 18](#_Toc165960933)

[2.1 Uczenie nadzorowane 20](#_Toc165960934)

[2.1.1 Regresja Logistyczna 21](#_Toc165960935)

[2.1.2 Metoda k najbliższych sąsiadów 23](#_Toc165960936)

[2.1.3 Drzewo Decyzyjne 24](#_Toc165960937)

[2.2 Uczenie nienadzorowane 26](#_Toc165960938)

[2.2.1 Sztuczne sieci neuronowe 27](#_Toc165960939)

[2.2.2 Deep Learning 28](#_Toc165960940)

[2.2.3 Sieci neuronowe konwolucyjne (CNNs) 32](#_Toc165960941)

[2.2.4 Rekurencyjne sieci neuronowe (RNNs) 32](#_Toc165960942)

[2.3 Algorytmy Przetwarzania Języka Naturalnego 40](#_Toc165960943)

[2.3.1 Analiza składowa i sematyczna 40](#_Toc165960944)

[2.3.2 Modelowanie tematyczne i klasyfikacja tekstu 40](#_Toc165960945)

[2.3.3 Techniki reprezentacji danych tekstowych 40](#_Toc165960946)

[3 Analiza sentymentu w mediach społecznościowych 40](#_Toc165960947)

[3.1 Charakterystyka i przygotowanie danych 40](#_Toc165960948)

[3.2 Metody redukcji wymiarów w NLP 41](#_Toc165960949)

[3.3 Analiza sentymentu z użyciem VADER i NLTK 41](#_Toc165960950)

[3.4 Budowa i optymalizacja modeli klasyfikacyjnych 41](#_Toc165960951)

[[dodać połączenie między regularyzacją, a optymalizacją] 46](#_Toc165960952)

[3.4 Wizualizacja danych i wyników Interpretacja i dyskusja nad wynikami analizy sentymentu 51](#_Toc165960953)

[3.5 Wymagania techniczne i sprzętowe 51](#_Toc165960954)

[Zakończenie 52](#_Toc165960955)

[Bibliografia 52](#_Toc165960956)

[Spis rysunków 54](#_Toc165960957)

[Spis tabel 55](#_Toc165960958)

[Załączniki 55](#_Toc165960959)

[Streszczenie 55](#_Toc165960960)

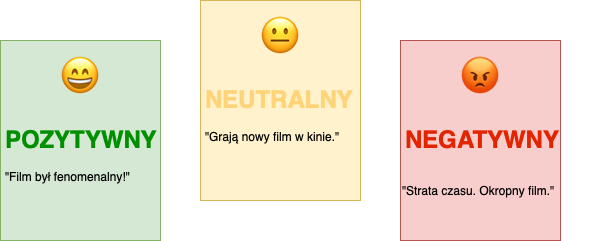
# Wstęp

TBC

# 1 Analiza sentymentu

## 1.1 Wprowadzenie do Analizy sentymentu

Analiza sentymentu (SA, od 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 treści, co jest niezmiernie istotne w kontekście szybko zmieniających się trendów i dynamiki społecznej. 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.

A diagram of different colored boxes

Description automatically generated

Rysunek 2. Schemat procesu reagowania na problemy z nową wersją aplikacji, 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 i zrozumienie subtelnych zmian w percepcji marki, produkcie czy usłudze, co jest nieocenione dla decydentów biznesowych. Wykorzystanie tej technologii do monitorowania i reagowania na opinie klientów w czasie bliskim do rzeczywistego będzie stanowić kluczowy czynnik różnicujący w konkurencyjnym środowisku rynkowym. 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 oraz przekonania wyrażane przez swoich konsumentów, zyskują cenne wskazówki, które mogą przekuć w działania prowadzące do wzrostu lojalności kupujących, poprawy reputacji marki, jak również w konsekwencji zwiększenia przychodów. W erze cyfrowej, gdzie dane stanowią nową walutę, analiza sentymentu w mediach społecznościowych, takich jak Twitter, ujawnia się jako fascynujące pole badań naukowych i potężne narzędzie biznesowe. Mikroblogowanie stało się popularnym narzędziem komunikacyjnym wśród użytkowników Internetu, gdzie miliony osób codziennie dzielą się opiniami na różne aspekty życia. Strony mikroblogowe są bogatym źródłem danych do analizy opinii i analizy sentymentu. (Pak & Paroubek, 2010). Portale społecznościowe stały się przez to nieocenionym miejscem pozyskiwania informacji do badania zachowań, reakcji, opinii użytkowników, aby móc im zaoferować doskonalsze, spersonalizowane usługi, które sprostają ich wymaganiom. Warto zauważyć, że z wnioskowania po treści komentarzy korzystają aktualnie nie tylko branża IT. Wiele marek przeniosło swoje sklepy do Internetu, ale także stara się tworzyć sieć społeczną wokół swoich produktów w 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.

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.

A screenshot of a computer

Description automatically generated

Tabela 1. Złożoność analizy sentymentu w różnych typach zdań, Źródło: opracowanie własne

W niniejszej pracy dyplomowej dokonana zostanie dogłębna analiza metod i technik stosowanych w analizie sentymentu, z naciskiem na ich zastosowanie w różnych obszarach biznesowych oraz wyzwania związane z interpretacją danych tekstowych. Praca ta ma na celu nie tylko zbadanie istniejących narzędzi i metod, ale również identyfikację potencjalnych obszarów do dalszych badań i rozwoju w tej dynamicznie rozwijającej się dziedzinie.

Analiza sentymentu, pierwotnie zakorzeniona w lingwistyce i informatyce, zyskała znaczący moment do rozwoju dzięki dynamicznym postępom w dziedzinie nauki o danych. Jako integralna część uczenia maszynowego, przetwarzanie języka naturalnego (NLP) doświadczyło znaczących innowacji i usprawnień. Analiza sentymentu przeszła transformację, stając się bardziej zaawansowaną oraz efektywną technologią. Przez wiele lat była ona blokowana przez brak wystarczającej ilości danych, zbyt słabe 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 technikach, bazujących na leksykonach zawierających słowa oznaczone emocjami. Jednakże, to właśnie szybki rozwój w dziedzinie algorytmiki i adaptacyjnych modeli uczenia maszynowego, w tym uczenia głębokiego (Deep Learning), spowodował, że analiza sentymentu przekształciła się w wysoce precyzyjne narzędzie, zdolne do dokładnych i wielowymiarowych analiz emocjonalnych zawartości tekstowej. Zdolność do precyzyjnego rozpoznawania oraz interpretacji złożonych wzorców zachowań w wypowiedziach jest nie tylko triumfem technologicznym, ale również stanowi nieocenioną wartość przez samodzielną umiejętność wykrywania wzorów zachowań. W obecnym trendzie technologicznym, gdzie dane stanowią kluczowy zasób, zastosowanie zaawansowanych technik otwiera przed przedsiębiorstwami nowe horyzonty strategiczne, a przed naukowcami wiele obszarów eksploracji. Patrząc na tę kwestię od strony biznesowej, firmy wykorzystujące te innowacje mogą zyskać głęboki wgląd w percepcje, jak również nastroje klientów, co jest nieocenione w procesie optymalizacji strategii marketingowych, rozwoju produktów czy usług. 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

Zatrudnienie jednostek zajmujących się analizą sentymentu, umożliwi nie tylko usprawnienie dotychczasowych rozwiązań, umożliwiając przedsiębiorstwom reagowanie na bieżące trendy i nastroje. Wdrożenie tego rozwiązania umożliwi także zaoszczędzenie funduszy w procesie budowania nowych rozwiązań, ponieważ będą już znane kierunki warte obrania, a także antycypowanie przyszłych zmian i dostosowywanie się do nich w sposób proaktywny. Dzięki temu, firmy nie tylko reagują na rynek, ale również kształtują go, będąc o krok przed konkurencją i w pełni wykorzystując potencjał ukryty w danych.

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 recenzjach 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. „tweedów” dotyczących usług doręczeniowych, 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 Twitter, oczyszczenia pobranych danych, analizy subiektywnej, budowy modelu ontologii oraz analizy sentymentu, naukowcy byli w stanie identyfikować i przetwarzać negatywne sentymenty związane z tweet’ami (Thakor & Sasi, 2015). Jak widać na rysunku 4 w procesie budowy modelu, z danych zostały usunięte hiperłącza do stron internetowych oraz znaki specjalne, które mogłyby utrudniać interpretację. Wykorzystano skrypty napisane w języku programowania *Python* do automatycznego pozyskania i oczyszczenia treści co było kluczowe w dalszym procesie badawczym. Kolejnym krokiem było zastosowanie metod analizy językowej wykorzystując biblioteki NLP w oprogramowaniu GATE, które umożliwiło identyfikację i adnotację rzeczowników oraz czasowników. Korzystając z paradygmatu programowania obiektowego rzeczowniki zdefiniowano jako obiekty, a czasowniki jako właściwości tych obiektów w kontekście ontologii. Informacje te zostały wykorzystane do budowy modelu za pomocą oprogramowania *Protégé*.

A screenshot of a cell phone

Description automatically generated

Rysunek 4. Proces Analizy Sentymentu Oparty na Ontologii dla Treści Mediów Społecznościowych, Źródło: opracowanie własne

W tym modelu, klasy, obiekty i właściwości obiektów zostały określone odpowiednio jako encje, właściwości obiektów. 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 administrowanie 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 wykorzystania de facto gramatyki i stylów pisowni. SentiStrength został zastosowany do analizy komentarzy na portalu interentowym „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 pozytywnego i negatywnego sentymentu, co było nowatorskie w porównaniu do tradycyjnych metod analizy sentymentu, które zazwyczaj koncentrowały się tylko na pozytywnych lub negatywnych emocjach, a nie na ich sile. Przykład tej pracy pokazuje, że jak wiele płaszczyzn należy uwzględnić, aby poprawnie zaadresować przekaz wypowiedzi i emocje mu towarzyszące. Te przykłady ilustrują, że analiza sentymentu wielorako przekształca dane tekstowe w cenne wskazówki, które firmy mogą wykorzystać do przewidywania trendów, personalizacji ofert i kształtowania strategii. W erze cyfrowej, gdzie dane są nowym surowcem, efektywne wykorzystanie tej techniki staje się kluczem do sukcesu.

## 1.2 Podstawy Przetwarzania Języka Naturalnego (NLP)

Analiza sentymentu jest częścią Przetwarzania Języka Naturalnego (NLP, od ang. Natural Language Processing), które skupia się na interakcjach między językiem ludzkim a komputerami. Znajduje się na przecięciu informatyki, sztucznej inteligencji i lingwistyki obliczeniowej (Lopez & Kalita, 2017). 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, Zhao Cai, Chee-Wee Tan, Qian Huang i Hefu Liu 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. Artykuły, które adresowały zagadnienia z NLP pochodziły z wielu dziedzin nauki (Rysunek 5). Wykazane przez nich wyniki wskazują, że w kolejnych latach technika ta będzie wzrastać na znaczeniu w biznesie i zarządzaniu.

A pie chart with text on it

Description automatically generated

Rysunek 5. 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.

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, firmy 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. 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. Dzięki NLP, 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, 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 istotne 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 różnych tła językowego. Aby dojść do definicji tokena, który jest zarówno lingwistycznie znaczący, jak i metodologicznie użyteczny (Kit & Webster, 1992).

A screenshot of a computer

Description automatically generated

Rysunek 6. 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. Stemming jest często postrzegany jako narzędzie zwiększające recall, czyli zdolność do odzyskiwania informacji. W językach o stosunkowo prostej morfologii wpływ stemming’u 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.

A screenshot of a computer

Description automatically generated

Rysunek 7. 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, 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 to kanoniczna forma 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.

A black and white rectangular object with a green rectangle in the middle

Description automatically generated

Rysunek 8. 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. Dzięki temu mogą one lepiej zrozumieć zachowania i preferencje użytkowników, co pozwala na optymalizację produktów i usług oraz tworzenie bardziej spersonalizowanych doświadczeń dla klientów.

W epoce cyfrowej, gdzie dane tekstowe są nieustannie generowane i przetwarzane, umiejętność efektywnego wykorzystania technik NLP staje się kluczowa dla firm technologicznych. Pozwala im to nie tylko na uzyskanie przewagi konkurencyjnej, ale także na podejmowanie świadomych i strategicznych decyzji biznesowych opartych na dogłębnej analizie danych.

Te techniki są fundamentem dla bardziej zaawansowanych operacji w NLP, umożliwiając efektywne przetwarzanie tekstu, co jest kluczowe w analizie sentymentu. Poprzez ich zastosowanie, systemy NLP mogą lepiej radzić sobie z różnorodnością i złożonością języka naturalnego, co przekłada się na dokładniejszą i bardziej wiarygodną analizę sentymentu. 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 w handlu akcjami; upraszczanie interakcji z klientami za pomocą chatbotów na platformach handlowych, czyniąc interakcję bardziej przyjemną; analizowanie problemów obywateli z dużych zbiorów danych 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 głębokie uczenie się, może zapewnić jeszcze dokładniejsze wyniki odpowiadające na potrzeby użytkowników (Wu i Mircea, 2021).

## 1.3 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 (NLTK), 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, protokołów i innych małych języków tekstowych. Wyrażenia regularne pojawiły się w kontekście teorii języków formalnych, a ich podstawowe zastosowanie to część skanerów 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.

CountVectorizer, często stosowany w bezpłatnej bibliotece oprogramowania do języka Python scikit-learn, służy do konwersji kolekcji dokumentów tekstowych na macierz liczb całkowitych, reprezentujących częstotliwość występowania słów. 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.

A screenshot of a computer

Description automatically generated

Rysunek 9. Przykłady użycia CountVectorizer dla różnych dokumentów, Źródło: opracowanie własne

NLTK to z kolei wszechstronna platforma do budowania programów języka Python do pracy z danymi języka ludzkiego, oferująca łatwe w użyciu interfejsy do korpusów i zasobów leksykalnych, jak również narzędzia do klasyfikacji, tokenizacji, stemmingu, tagowania i parsowania. W analizie sentymentu, NLTK pozwala na szczegółową i złożoną analizę języka naturalnego, co ma istotne znaczenie dla firm technologicznych w analizie danych klientów, mediów społecznościowych czy treści online.

Wykorzystanie tych narzędzi w analizie sentymentu otwiera przedsiębiorstwom technologicznym możliwość nie tylko efektywnego przetwarzania i analizowania danych tekstowych, ale także przekształcania surowych danych w cenne wglądy biznesowe. W dobie cyfrowej, gdzie analiza danych tekstowych odgrywa kluczową rolę w zrozumieniu rynku i potrzeb klientów, narzędzia takie jak Regex, CountVectorizer i NLTK stanowią niezbędny element w budowaniu przewagi konkurencyjnej na rynku.

W dziedzinie analizy sentymentu, znaczący postęp dokonany został dzięki bogatej literaturze naukowej i innowacyjnym rozwiązaniom technologicznym. Rozważając kluczowe prace i badania, nasuwa się obserwacja, że rozwój tej dziedziny jest ściśle powiązany z postępami w uczeniu maszynowym i przetwarzaniu języka naturalnego.

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, co ma bezpośrednie zastosowanie w monitorowaniu opinii konsumentów, analizie trendów rynkowych oraz zarządzaniu reputacją marki.

Porównując istniejące rozwiązania i technologie, 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. Co istotne, ich aplikacja w świecie biznesu otwiera drogę do głębszego zrozumienia potrzeb klientów, umożliwiając firmom tworzenie bardziej spersonalizowanych strategii marketingowych i komunikacyjnych.

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.

Podsumowując, przegląd literatury i istniejących rozwiązań w dziedzinie analizy sentymentu wskazuje na dynamiczny rozwój tej branży, który jest napędzany zarówno przez postępy akademickie, jak i innowacje technologiczne. 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 biznesowej.

## 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 kompleksowość języka naturalnego, który charakteryzuje się nie tylko bogactwem ekspresji, ale także ironią, sarkazmem i zmiennymi kontekstami kulturowymi. Analiza sentymentu jest zadaniem subiektywnym i dlatego nasze dane mogą nadal cierpieć na uprzedzenia etykietowania, które dotykają większość 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. Nowe wyrażenia i ciągłe zmiany w strukturze języka wymagają od systemów analizy sentymentu elastyczności i adaptacyjności. Stały rozwój algorytmów Uczenia Maszynowego, które potrafią śledzić i dostosowywać się do tych zmian, jest kluczowy dla efektywnej Analizy Sentymentu.

Dodatkowo, przetwarzanie i analiza ogromnych ilości danych generowanych przez media społecznościowe wymaga zaawansowanej infrastruktury technologicznej oraz efektywnych metod przetwarzania danych. Wyzwanie to dotyczy zarządzania i analizowania dużych zbiorów danych w sposób wydajny i skuteczny.

Organizacje działają w dynamicznych i ciągle zmieniających się środowiskach, składających się z aktorów takich jak konkurenci i rząd, którzy mają wpływ na to, jak organizacja może i powinna prowadzić biznes. To z kolei wywiera różne rodzaje presji na zdolność i skłonność organizacji do przyjęcia sztucznej inteligencji. (Enholm, Papagiannidis, Mikalef, & Krogstie, 2021). W kontekście biznesowym, te wyzwania przekładają się na potrzebę inwestycji w nowoczesne technologie i specjalistyczną wiedzę. Firmy, które chcą efektywnie wykorzystać analizę sentymentu, muszą podejmować świadome decyzje dotyczące wyboru narzędzi i technologii zdolnych do radzenia sobie z wymienionymi problemami. Sukces w tej dziedzinie może prowadzić do głębszego zrozumienia potrzeb klientów, efektywniejszego reagowania na zmieniające się tendencje rynkowe oraz lepszego zarządzania reputacją marki.

Podsumowując, chociaż analiza sentymentu oferuje obiecujące możliwości dla biznesu, wymaga ona jednocześnie skrupulatnego podejścia do istniejących wyzwań. Rozwiązanie tych problemów za pomocą innowacyjnych metod i technologii jest kluczowe dla efektywnego wykorzystania analizy sentymentu jako narzędzia do uzyskiwania przewagi konkurencyjnej na współczesnym rynku.

# 2 Fundamenty teoretyczne metod klasyfikacji emocjonalnej tekstu

Analiza emocjonalna tekstu, znana również jako analiza sentymentu, wykorzystuje złożone algorytmy do oceny i klasyfikacji informacji znajdujących się w tekstach zadanych jako dane wejściowe. Rozdział ten ma za zadanie teoretyczne wprowadzenie do algorytmiki, co pozwoli lepiej zrozumieć ich dalsze, praktyczne wykorzystanie w stworzonej aplikacji. Wiele rozwiązań systemów analizy sentymentu jest budowanych wokół klasyfikacji emocjonalnej, aby zidentyfikować oraz analizować subiektywnie kluczowe aspekty sentymentu.

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 i generalizować posiadane dane. 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 obiektywność 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. Taksonomia 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.

W tej pracy skoncentruję się na podejściu ML ze względu na jego adaptacyjność i moce predykcyjne do zmieniających się danych wejściowych lub testowych. 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. Warto zauważyć, że to właśnie dzięki tej zdolności do treningu na wskazanych przykładach, systemy te cechuje lepsza umiejętność generalizacji, co jest niezwykle wartościowe w szybko zmieniających się kontekstach, takich jak media społecznościowe. Kolejnym aspektem wartym poruszenia, jest kwestia, że uczenie maszynowe daje w tym momencie możliwość automatycznej 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 agentowi precyzyjnej miary jego błędu (bezpośrednio porównywalnej z wartościami wyjściowymi) (Bonaccorso, 2017). Jest to proces uczenia na podstawie zestawu danych zawierającego obserwacje wraz z określonymi etykietami (w tym przypadku sentymenty), na których model uczy się wykrywać schematy i w ten sposób może przewidzieć sentymenty dla nowych, nieokreślonych danych.

Jest wiele zastosowań, gdzie uczenie nadzorowane umożliwia automatyczna oraz skuteczną analizę dużych ilości tekstów, dostarczając opiniotwórczych informacji dla biznesu. To uczenie jest stosowane w aplikacjach, gdzie dane historyczne przewidują prawdopodobne przyszłe wydarzenia. Istnieje wiele praktycznych przykładów tego uczenia, na przykład aplikacja, która przewiduje gatunek irysa na podstawie zestawu pomiarów jego kwiatu. (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 listach. Przez lata zauważono wyjątkowy potencjał tego rozwiązania i poszerzono użytkowania o na przykład, medycynę, gdzie algorytmy pomagają w identyfikacji zmian nowotworowych w obrazach wyników badań medycznych. W finansach znalazło zastosowanie w przewidywaniu trendów na giełdzie papierów wartościowych, analizie ryzyka kredytowego czy automatyzacji doradztwa inwestycyjnego. Jednak ewolucja w tej dziedzinie ciągle postępuje i odkrywane są kolejne przeznaczenia, m. in. w uczeniu głęboki, 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 aspektów, których ważność tylko wzrasta, na przykład detekcja fałszywych wiadomości i dezinformacji. Klasyfikacja artykułów, tekstów pod kątem ich wiarygodności może obejmować analizę sentymentu w celu zidentyfikowania manipulacji czy fałszywych informacji. Ponieważ szerokie rozprzestrzenianie się fałszywych informacji może mieć poważny negatywny wpływ na jednostki i społeczeństwo, brak 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 zapewnia możliwość dostosowania dla danych z wieloma zmiennymi zależnymi. Sprawia to, że regresja logistyczna jest szczególnie użyteczna do analizy danych obserwacyjnych, gdy konieczne jest dostosowanie w celu zmniejszenia potencjalnego błędu wynikającego z różnic w porównywanych grupach (LaValley, 2008).

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

Funkcja sigmoidalna jest określona wzorem:

Gdzie:

e - to podstawa logarytmu naturalnego

x - jest zmienną niezależną

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 i 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 Najbliższego Sąsiada (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). Klasyfikacja metodą k najbliższych sąsiadów (k-NN) to proces, który polega na identyfikacji k obserwacji z zestawu treningowego, które są najbliższe nowej, nieznanej obserwacji, na podstawie wybranej metryki odległości. Służą do tego podejścia takie jak odległość euklidesowa, Manhattan czy Minkowski. 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.

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.

Wyraźną zaletą tego algorytmu jest prostota implementacji i intuicyjność działania, wynikająca z podobieństw płynących z życia codziennego. 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. 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).

### 2.1.3 Drzewo Decyzyjne

Drzewa decyzyjne (ang. Decision Tree) są graficzną metodą wspomagania procesu decyzyjnego. 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). Prostota i intuicyjność, prowadzą do łatwości w przedstawieniu nawet dla osób, które nie są ekspertami w dziedzinie analizy danych. 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. Procedurę tę 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 one faktycznie grafami skierowanymi i acyklicznymi, 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 i zestawie 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 podejmowania 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 interpretacji 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 (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 nienadzorowane

Uczenie nienadzorowane (ang. Unsupervised Learning) ułatwia analizę surowych zbiorów danych, pomagając tym samym w generowaniu analitycznych wniosków z danych bez wyznaczonych etykiet (Usama et al., 2019). Jedną z głównych różnic między uczeniem maszynowym nienadzorowanym a uczeniem maszynowym nadzorowanym jest brak zestawu treningowego dla tego pierwszego i w związku z tym, brak oczywistej roli dla walidacji krzyżowej. Drugą ważną różnicą jest to, że chociaż większość algorytmów klastrowania jest formułowana w terminach kryterium optymalności, zazwyczaj nie ma gwarancji, że uzyskano rozwiązanie optymalne globalnie. (Gentleman & Carey, 2008).

Uczenie nienadzorowane posiada swoją mocną stronę w identyfikacji nieznanych wcześniej wzorców i struktur danych. Umożliwia to odnajdywanie zależności i grupowania, które mogą nie być trudne do odnalezienia dla człowieka. Przy dużych zbiorach danych, daje to ewidentną przewagę, gdy dane nie posiadają etykiet. Przez to techniki te odnalazły zastosowanie w wielu dziedzinach. Grupowanie podobnych do siebie obserwacji, ma swoje przeznaczenie w identyfikacji segmentów klientów w marketingu, typach zachowań użytkowników w serwisach internetowych czy w analizie danych genetycznych. Redukcja wymiarowości ułatwia wizualizację danych wysokowymiarowych jest pomocna w poprawie wydajności innych modeli uczenia maszynowego przez zmniejszenie liczby cech. W procesie wykrywania anomalii uczenie nienadzorowane identyfikuje dane, które są obserwacją odstającą, anomalie, co ma kluczowe znaczenie w wykrywaniu oszustw finansowych, monitorowaniu stanu maszyn w produkcji czy w cyberbezpieczeństwie. Przez wnioskowanie asocjacyjne można odkrywać reguły, które opisują silne zależności między zmiennymi, przydatne w analizie koszykowej i systemach rekomendacyjnych, które mają swoje zastosowanie w e-commerce czy handlu detalicznym.

### 2.2.1 Sztuczne sieci neuronowe

Sztuczne sieci neuronowe (ANN od ang. Artificial Neural Network) 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 generowanie 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 Deep Learning

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). 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 warstawy 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 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 11. 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 12. 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, samoprowadzące 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. Warto jednak pamiętać, że rozwój ten jest związany z wyznwami, np. Z zasobożernością, ponieważ do treningu głębokich sieci neuronowych potrzebne jest posiadanie specjalistycznego sprzędty oraz mocy obluiczeniowej. Automatyczność jest ograniczeniem dla dziedziń takich jak prawo czy medycyna, gdzie oczekuje się weryikowalości I odpoweidzialosci, przez to w tych branzach modele te dą krytyykowane. 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.

### 2.2.3 Sieci neuronowe konwolucyjne (CNNs)

Sieci neuronowe konwolucyjne (CNN od ang. Convolution Neural Networks) są analogiczne do tradycyjnych sieci ANNs, 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, jednak nie zostaną one użyte w tej pracy podczas analizy sentymentu.

### 2.2.4 Rekurencyjne sieci neuronowe (RNNs)

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

A diagram of a complex network

Description automatically generated

Rysunek 13. 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. Koncept architektury rekurencyjnych sieci neuronowych (RNN) jest głęboko zakorzeniony w przełomowej pracy Davida Rumelharta z 1986 roku, ponieważ to tam został wprowadzony przez niego aspekt pętli wewnętrznych w architekturze sieci, co dało możliwość przetwarzania sekwencyjnych danych w czasie, predysponując sieci do wykorzystania informacji z poprzednich kroków czasowych do wpływania na przetwarzanie bieżących i przyszłych danych. RNN składają się z ukrytych stanów o wysokiej wymiarowości z nieliniową dynamiką (Sutskever, Martens, & Hinton, 2011). Struktura ukrytych stanów działa jako pamięć sieci, a stan warstwy ukrytej w danym momencie jest uzależniony od jej poprzedniego stanu (Mikolov, Joulin, Chopra, Mathieu, & Ranzato, 2014). Ta struktura umożliwia RNN przechowywanie, zapamiętywanie i przetwarzanie złożonych sygnałów z przeszłości przez długie okresy czasu. RNN mogą mapować sekwencję wejściową na sekwencję wyjściową w bieżącym kroku czasowym i przewidywać sekwencję w następnym kroku czasowym (Salehinejad, Sankar, Barfett, Colak, & Valaee, 2018). Jest to rozwinięcie podstawowych sieci neuronowych przekazujących sygnał wprzód (feed-forward), innymi słowy, sieci RNN w co najmniej jednej z ich warstw neuronowych, są wzbogacane o tzw. „pętle sprzężenia zwrotnego”. W RNN, w przeciwieństwie do sieci typu feed-forward, informacja nie jest przekazywana jednokierunkowo, może ona natomiast obiegać sieć wewnątrz poprzez pętle sprzężenia zwrotnego. Pętle te umożliwiają sieci, aby każdy kolejny element sekwencji był przetwarzany z uwzględnieniem nie tylko aktualnych informacji wejściowych, ale zarówno kontekstu dostarczonego przez uprzednio przetworzone elementy. W realnym świecie zapewnia to możliwość zrozumienia gramatyki, znaczenia całego zdania, akapitu w analizie języka naturalnego. Aspektem, który warto poruszyć jest trenowanie rekurencyjnych sieci neuronowych, które jest jednocześnie wyzwaniem, ale nieodłącznym krokiem użytkowania ich. Ogólnie rzecz biorąc, wraz z ulepszonymi architekturami wyjaśnionymi poniżej, szybkie implementacje i lepsza heurystyka podążania za gradientem sprawiły, że szkolenie RNN stało się wykonalne (Lipton, Berkowitz, & Elkan, 2015). Proces trenowania zaczyna się w fazie propagacji wprzód (ang. Forward Propagation), gdzie zostają przetwarzane dane wejściowe, przesyłając informacje przez sieć w kierunku od warstw wejściowych do wyjściowych. Potem na końcu sekwencji lub w wyznaczonych punktach następuje obliczanie błędu, poprzez obliczanie różnicy między przewidywanym, a rzeczywistym wyjściem, używając funkcji straty, np. błędu średniokwadratowego. Kolejnym punktem w działaniu algorytmu jest propagacja wsteczna przez czas (ang. Backpropagation Through Time - BPTT), która poszerza działanie standardowej propagacji wstecznej w RNN w czasie, tworząc szereg połączonych ze sobą sieci wyprzedzających (ang. feedforward networks). Każdy krok czasowy odpowiada jednej warstwie w tej rozwiniętej sieci, a wagi pomiędzy warstwami są dzielone pomiędzy krokami czasowymi. Rozwiniętą sieć można traktować jako bardzo głęboką sieć wyprzedzającą, w której wagi są dzielone pomiędzy warstwy. Pomimo wielu unikalnych zalet RNN mierzy się z trudnościami w opisanych uprzednio procesie trenowania. Podczas BPTT, gradienty mogą zanikać (stawać się coraz mniejsze) co ma swoje następstwa w pojawiających się trudnościach nauki zależności na długich sekwencjach lub gradienty mogą też eksplodować (rosnąć nieproporcjonalnie), co może prowadzić do trudności numerycznych (Sherstinsky, 2020). Kolejnym minusem jest zasobożerność przez złożoność obliczeniową BPTT. Podsumowując, trenowanie RNN jest złożonym, jednak bezwarunkowo niezbędnym procesem w dziedzinie uczenia maszynowego, oferującym imponujące narzędzia do analizy i generowania danych sekwencyjnych.

Pod koniec lat 90. XX wieku praca S. Hochreitera i J. Schmidhubera wniosła następny przełom, przedstawiając koncepcję komórki z długą pamięcią krótkotrwałą (ang. Long Short-Term Memory network, w skrócie LSTM). To te badania pokazały, że LSTM w sposób znaczący poprawia zdolność modeli do uczenia się długoterminowych zależności w danych. Było to wcześniej dużym wyzwaniem dla wcześniejszych wersji RNN, a dzięki tej innowacji zwiększono dokładność, efektywność w przetwarzaniu informacji oraz zapamiętywaniu danych przez dłuższe przedziały czasowe, co w kolejnym kroku umożliwiło rozwiązywanie złożonych problemów związanych z przetwarzaniem języka naturalnego czy rozpoznawaniem mowy. W porównaniu z uczeniem rekurencyjnym w czasie rzeczywistym, propagacją wsteczną w czasie, korelacją kaskadową rekurencyjną, sieciami Elmana i fragmentacją sekwencji neuronowych, LSTM prowadzi do znacznie większej liczby udanych przebiegów i uczy się znacznie szybciej. LSTM rozwiązuje także złożone, sztuczne zadania o długim opóźnieniu, które nigdy nie zostały rozwiązane przez poprzednie, rekurencyjne algorytmy sieciowe (Hochreiter & Schmidhuber, 1997).

A diagram of a cell block

Description automatically generated

Rysunek 14. 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. (1997). 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 14 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 połaczenia 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 15. 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 LSTM służy do przetwarzania danych sekwencyjnych, takich jak serie czasowe, dane nagrań audio, czy też tekstu. Składa się ona z kilku fundamentalnych elementów, które współpracując podejmują decyzję o sposobie przechowywania informacji, aktualizacji czy przekazywania ich. W oryginalnej architekturze LSTM bez bramki zapominania składa się ona z dwóch elementów: bramki wejściowej i wyjściowej. W tym zamyśle pierwszym elementem komórki LSTM jest bramka wejściowa (ang. input gate), która podczas aktualizacji stanu komórki może decydować jakie nowe informacje mogą być przechowywane w stanie komórki, oceniając znaczenie nowych informacji. Składają się na wykonanie tego procesu dwie funkcje aktywacji: sigmoidalna i tanh. Funkcja sigmoidalna odpowiada za przechowywanie/ odrzucanie informacji, natomiast funkcja tanh polega na odejmowaniu lub dodaniu nowych informacji od stanu komórki. Stan komórki (ang. cell state) to fundamentalny nośnik informacji w komórce LSTM. Stan ten przechowuje zarówno krótko- jak i długoterminowe informacje, dokonywane jest to poprzez aktualizowanie. Stan komórki jest aktualizowany następująco: następuje pomnożenie starego stanu przez wartość z bramki zapominania i dodanie nowych wartości kandydatów po wcześniejszej filtracji w bramce wejściowej. Kolejna bramka jest nazywana wyjściową (ang. output gate), decyduje ona jakie informacje mogą być wysyłane na podstawie stanu komórki. Stanowi ona mechanizm umożliwiający sieci nie tylko zachowanie ważnych informacji przez długi czas, ale także odpowiednie reagowanie na bieżące dane wejściowe poprzez kontrolowane przekazywanie informacji ze stanu komórki do wyjścia sieci. Określa ona, które informacje ze stanu komórki powinny zostać przekazane dalej poprzez zastosowanie funkcji aktywacji (najczęściej funkcji sigmoidalnej) w tworzeniu kombinacji aktualnych danych wejściowych oraz poprzedniego stanu ukrytego. Funkcja aktywacji generuje wartości, które są utożsamiane ze stopniem w jakim poszczególne elementy stanu komórki wypływają na stan ukryty wyjściowy. Ostatecznie wynik jest normalizowany poprzez funkcję aktywacji tanh, gdzie wynik jest łączony z przefiltrowanym stanem komórki. Bramka ta ma swoje kluczowe znaczenie w selektywnym przekazywaniu tylko relatywnie ważnych informacji.

A diagram of a computer system

Description automatically generated

Rysunek 16. 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), która decyduje o tym jakie informacje z poprzedniego stanu komórki będzie zapomniana czy też nastąpi jej zachowanie. Jest to dokonywane poprzez zastosowanie sigmoidalnej funkcji aktywacji. Funkcja ta na wejściu przyjmuje aktualne dane wraz ze stanem ukrytym generując wartości w zakresie 0 do 1, wartości te odpowiednio oznaczają zapomnienie i zachowanie informacji. Dzięki opisanym właściwościom LSTM może reagować na zmiany w danych wejściowych jednocześnie zachowując istotną wiedzę w przeszłości. 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. W przypadku problemów z długim opóźnieniem, […] LSTM radzi sobie z szumem, reprezentacjami rozproszonymi i wartościami ciągłymi. W przeciwieństwie do automatów skończonych lub ukrytych modeli Markowa, LSTM nie wymaga apriorycznego wyboru skończonej liczby stanów. W zasadzie może obsłużyć nieograniczone numery stanowe (Hochreiter & Schmidhuber, 1997). Pomimo zalet sieci neuronowe LSTM są powolne ze względu na dużą równoległość i sekwencyjny charakter. Miejmy nadzieję, że problem ten zostanie rozwiązany poprzez zaprezentowanie w najbliższej przyszłości niezawodnego akceleratora sprzętowego (Smagulova & James, 2019).

Następną innowacją w dziedzinie nauki o RNN było wprowadzenie w 2014 roku koncepcji bramek rekurencyjnych (ang. Gated Recurrent Unit, w skrócie: GRU). W tym badaniu Cho i inni pokazują, że neuronowe tłumaczenie maszynowe sprawdza się stosunkowo dobrze w przypadku krótkich zdań niezawierających nieznanych słów, ale jego wydajność gwałtownie spada wraz ze wzrostem długości zdania i liczby nieznanych słów. Co więcej, odkrywają, że proponowana bramkowana rekurencyjna sieć splotowa automatycznie (ang. gated recursive convolutional network) uczy się struktury gramatycznej zdania (Cho, van Merriënboer, Bahdanau, & Bengio, 2014).

A diagram of a computer system

Description automatically generated

Rysunek 17. 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.

Podczas tych badań zostało odkryte, że jest możliwe uproszczenie architektury LSTM, struktury modelu umożliwia otrzymanie porównywalnej, a nawet w niektórych scenariuszach lepszej wydajności, a jednocześnie zmniejsza złożoność obliczeniową. Celem tej innowacji jest, aby każda powtarzająca się jednostka adaptacyjnie przechwytywała zależności w różnych skalach czasowych. Podobnie jak jednostka LSTM, GRU posiada jednostki bramkujące, które modulują przepływ informacji wewnątrz jednostki, jednak bez posiadania osobnych komórek pamięci (Chung, Gulcehre, Cho, & Bengio, 2014). W przeciwieństwie do LSTM, GRU ma tylko dwie bramki (bramka resetująca i bramka odświeżająca) i nie posiada komórki pamięci (Tüske et al., 2016). Bramka aktualizacyjna (ang. update gate) ma za zadanie wyznaczyć, ile poprzednich informacji z uprzednich kroków należy przekazać do kolejnych iteracji. Decydując co należy zachować w pamięci, działa ona podobnie do bramki zapomnienia i bramki wejściowej w LSTM. Natomiast bramka resetująca (ang. reset gate) kontroluje, ile informacji z przeszłości należy zapomnieć, co jest kluczowe w modelowaniu zależności między odległymi punktami w sekwencji. Przepływ danych wejściowych i z poprzedniego stanu następuje przez obie bramki, które dynamicznie regulują przepływ informacji. To przetwarzanie umożliwia GRU uczenie się, które dane są istotne. Na podstawie wyjścia z obu bramek GRU tworzy nowy stan kandydata, który jest kombinacją przeszłych informacji (po uwzględnieniu bramki resetowania) oraz nowych informacji z bieżącego wejścia. W tej sposób obliczana jest bieżąca zawartość pamięci (ang. current memory content) na bramce resetowania i łączeniu przekształconych poprzedni stan ukryty i bieżące wejście. Wynik to przeszedł przez funkcję aktywacji stycznej hiperbolicznej do wygenerować aktywację kandydata (Mortezapour Shiri, Perumal, Mustapha, & Mohamed, 2023). Aby otrzymać ostateczny stan na teraźniejszy krok, model łączy poprzedni stan z nowym łącząc je z wykorzystaniem bramki aktualizacji.

Model GRU ma mniej bramek do aktualizacji i resetowania podczas obliczania stanu ukrytego

w porównaniu z modelem LSTM. Ta prostota sprawia, że GRU jest szybsze i bardziej wydajne

(Zegarra & Vargas-Machuca, 2023). Efekt tej architektury ma kluczowe znaczenie w dziedzinie sztucznej inteligencji z wielu powodów, między innymi: pozwala na przetwarzanie większych zasobów danych w krótszych czasie, umożliwia to zastosowanie takiego rozwiązania na urządzeniach o ograniczonej mocy obliczeniowej, zmniejszenie zużycia energii w procesie uczenia, poprawia skalowalność, co jest kluczowe w przetwarzaniu dużych wolumenów danych w czasie bliskim do rzeczywistego, a także warto wspomnieć zmniejszenie wymagań sprzętowych zwiększy dostępność rozwiązania w szerokich kręgach badaczy i praktyków, ponieważ nie wszyscy z nich mają dostęp do dużych centrów danych czy specjalistycznej infrastruktury. Warto jednak zauważyć, że to w LSTM będziemy mieć większą kontrolę, ze względu na większą ilość bramek. LSTM w teorii powinien zapamiętywać dłuższe sekwencje niż GRU lepiej i dokonywać trafniejsze wyniki badań dla zadań, które wymagają modelowania relacji, gdzie odległości są większe. Jednym ze scenariuszy takich sytuacji będzie tłumaczenie długich zdań, gdzie konieczne jest uwzględnienie kontekstu z początku zdania przy tłumaczeniu jego zakończenia. Jako szczegółowy przykład można przedstawić, kwestie formy czasownika w zdaniu, która zależy od płci podmiotu występującego na początku. Natomiast, GRU jest prostszym modelem, łatwiejszym do modyfikacji oraz przy mniejszej ilości danych treningowych będzie trenował się szybciej. Ze względu na prostą strukturę i mechanizm, GRU jest szeroko stosowany w przetwarzaniu języka naturalnego ze względu na jego wysoką wydajność (Ni & Cao, 2020).

## 2.3 Algorytmy Przetwarzania Języka Naturalnego

### 2.3.1 Analiza składowa i sematyczna

Mechanizmy uwagi i ich zastosowanie

Rozpoznawanie encji nazwanych (NER)

Analiza zależnościowa (Dependency Parsing)

### 2.3.2 Modelowanie tematyczne i klasyfikacja tekstu

Modelowanie tematyczne (LDA, LSI)

Klasyfikacja sentymentu z wykorzystaniem modeli głębokiego uczenia

### 2.3.3 Techniki reprezentacji danych tekstowych

Bag of Words i TF-IDF

Embeddingi (Word2Vec, GloVe, FastText)

Kontekstowe modele językow

# 3 Analiza sentymentu w mediach społecznościowych

Rozdział ten będzie poświęcony opisowi danych, metod, procedur wykorzystanych w projekcie analizy sentymentu. Elaboruje on nad przyczynami selekcji metod w ramach modelu uczenia głębokiego oraz dostarcza sukcesywnego opisu ich wykorzystania w etapach. Zostaje również dostarczony szczegółowy wgląd w dane, poczynając od źródła pozyskania, poprzez eksploracyjną analizę danych, aż po techniki ich przetwarzania wstępnego potrzebne do zasilenia sieci neuronowej. W finalnej części określone są wymagania technologiczne, potrzebne do uruchomienia projektu.

## 3.1 Charakterystyka i przygotowanie danych

Projekt: podział danych (test, train, valid);

Zbiór treningowy/ uczący/ estymacyjny: Używany do budowy modeli. Służy do estymacji wszystkich modeli/ kandydatów Przykład: 60 - 70% całego zbioru danych.

Zbiór walidacyjny: Służy do wstępnej oceny modeli i doboru hiperparametrów. Służy do wyboru jednego najlepszego modelu. Przykład: 15- 20% danych.

Zbiór testowy: Wykorzystywany do ostatecznej oceny modelu. Służy do nieobciążonej estymacji błędów tego jednego wybranego modelu. Przykład: Ostatnie 15 - 20% danych.

A diagram of a diagram

Description automatically generated

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

Omówienie metod wstępnego przetwarzania tekstu w NLTK (tokenizacja, usuwanie słów nieznaczących, stemming i lematyzacja)

Użycie CountVectorizer i TfidfVectorizer z pakietu sklearn do transformacji tekstu na wektory

## 3.2 Metody redukcji wymiarów w NLP

Metody SA, hyperparameter tunning Zastosowanie PCA i Truncated SVD do redukcji wymiarowości danych tekstowych

Analiza wpływu redukcji wymiarów na klasyfikację sentyment

## 3.3 Analiza sentymentu z użyciem VADER i NLTK

## 3.4 Budowa i optymalizacja modeli klasyfikacyjnych

[dodać dlaczego tu mówię o regularyzacji]

Podstawową ideą wszystkich metod regularyzacji jest ograniczenie przestrzeni możliwych rozwiązań (Poggio, Torre, & Koch, 1985). 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 (ang. overfitting). 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 tego właśnie 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 temrinu 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 (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:

* oznacza estymatory współczynników dla metody Lasso
* *arg min β* wskazuje, że poszukujemy wartości wektora β, które minimalizują funkcję kosztu
* to suma kwadratów reszt z podziałem przez 2, gdzie to obserwowane wartości zmiennej zależnej, a wartości zmiennych niezależnych
* to termin regularyzujący zastosowany w metodzie Lasso, który kara za wielkość bezwzględną współczynników regresji, promując model o mniejszej złożoności

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 grzebietowa (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:

* to estymator współczynników regresji grzbietowej
* Symbol *arg min β* wskazuje, że szukamy takich wartości wektora *β*, które minimalizują wyrażenie w nawiasach klamrowych
* to suma po wszystkich N obserwacjach
* to rzeczywiste wartości zmiennej zależnej
* *β0* to wyraz wolny, a to wartości zmiennych niezależnych
* to suma iloczynów wartości zmiennych niezależnych i odpowiadających im współczynników regresji
* to termin regularyzujący, który karze za duże wartości współczynników regresji

Regularyzacja L2 karze duże wartości współczynników przez co poprawia odporność modeli na przeuczenie, ale w przeciwieństwie do L1, nie prowadzi do całkowitego wyzerowania współczynników.

Dropout 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 eksponencjalnie 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 18, 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.

A screenshot of a game

Description automatically generated

Rysunek 18. 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).

## [dodać połączenie między regularyzacją, a optymalizacją]

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 19). 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 19. Współczynnik uczenia: mały vs optymalny vs duży, Źródło: Jordan (2018)

Optymalizacja może być osiągana na wiele sposobów. Optymalizację symulacyjną 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. Proces iteracyjny Adama przedstawia się następująco. (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).

Accuracy, Precision, and Recall; Confusion Matrix

Macierz błędów (confusion matrix), która jest narzędziem wykorzystywanym w ocenie wydajności modeli klasyfikacyjnych.

* Klasa rzeczywista (Actual Class): Kategorie, do których faktycznie należą przypadki/obserwacje.
* Przewidywana klasa (Predicted Class): Kategorie, do których model przewidział przynależność przypadków/obserwacji.

Table

Description automatically generated with medium confidence

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

Macierz składa się z czterech kwadrantów:

* True Positive (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.

* False Negative (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.

* False Positive (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.

* True Negative (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.

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

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

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

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

Macierz błędów oraz powiązanych z nią metryk ma 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.

TBC

## 3.4 Wizualizacja danych i wyników Interpretacja i dyskusja nad wynikami analizy sentymentu

TBC

## 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 do wizualizacji; Keras i TensorFlow do głębokiego uczenia; seqeval do oceny wydajności; oraz NumPy do algebry macierzy. Keras, funkcjonujący na platformie TensorFlow, pełni kluczową rolę w operowaniu modelami głębokiego uczenia, takimi jak CNN, RNN i LSTM. Uruchomienie aplikacji zmusza do zainstalowania wskazanych modułów jak 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 zadzwięczają 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 na dużą skalę, analizy predykcyjnej oraz obliczeń naukowych (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 pakietami, które mogłeś zainstalować (Bloice & Holzinger, 2016). W celu umożliwienia współpracy z innymi osobami w przyszłości, całość aplikacji jest dostępna na platformie GitHub.

# Zakończenie

TBC

# Bibliografia

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

Chopra, A., Prashar, A., & Sain, C. (2013). Natural Language Processing. 1(4).

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

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

Erwig, M., & Gopinath, R. (2012). Explanations for Regular Expressions. 7212, 394–408. https://doi.org/10.1007/978-3-642-28872-2\_27

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

Jivani, A. G. (2011a). A Comparative Study of Stemming Algorithms. 2.

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

Liu, B. (2012). Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers. https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf

Lopez, M. M., & Kalita, J. (2017). Deep Learning applied to NLP (arXiv:1703.03091). arXiv. http://arxiv.org/abs/1703.03091

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

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

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. 2(1–2), 20.

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

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

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

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

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

Wu, R. M. X., & Mircea, M. (2021). E-Business: Higher Education and Intelligence Applications. BoD – Books on Demand.

# Spis rysunków

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

Rysunek 2. Schemat procesu reagowania na problemy z nową wersją aplikacji, Źródło: opracowanie własne

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

Rysunek 4. Ontology-based Sentiment Analysis Process for Social Media Content. Procedia Computer Science, 199-207, Źródło: opracowanie własne

Rysunek 5. 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.

Rysunki 6-9. Przykłady narzędzi i algorytmów używanych w przetwarzaniu języka naturalnego Źródło: opracowanie własne

Rysunek 13. 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 14. Przykład sieci z ośmioma jednostkami wejściowymi, czterema jednostkami wyjściowymi i dwoma blokami komórek pamięci o rozmiarze 2. Źródło: Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1745.

Rysunek 15. 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 16. 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 17. 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 18. Zastosowanie metody Dropout w konwencjonalnej sieci neuronowej., Źródło: opracowanie własne

Rysunek 19. Współczynnik uczenia: mały vs optymalny vs duży, Źródło: Jordan (2018), https://www.jeremyjordan.me/nn-learning-rate/

# Spis tabel

Tabela 1. Złożoność analizy sentymentu w różnych typach zdań, Źródło: opracowanie własne

# Załączniki

# Streszczenie