SZKOŁA GŁÓWNA HANDLOWA W WARSZAWIE
WARSAW SCHOOL OF ECONOMICS

Studium ……

Kierunek/makrokierunek: ……
Specjalność:** ……
Forma studiów: ……

Maciej Sadkowski
MS107984

Tytuł pracy

Praca magisterska
napisana w instytucie informatyki i gospodarki cyfrowej
pod kierunkiem naukowym Mariusza Rafało

Warszawa 20…

*Zastosować właściwe

** W przypadku braku specjalności lub braku deklaracji o specjalności wiersz należy pominąć

1. Wstęp

1.1 Znaczenie leków dla ludzi

1.2 Wybranie odpowiedniego leku na wybraną dolegliwość – jak możemy szybko się dowiedzieć, który lek jest najlepszy

1.3 Przedstawienie i opis zbioru komentarzy pacjentów

1.4 Znaczenie ocen leków i zaproponowanie modelu do ich określenia

2. Opisanie NLP oraz użytych algorytmów do analizy NLP

2.1 Czym jest text mining i NLP?

Do zrozumienia analizy tekstu należy najpierw wiedzieć czym jest text mining czy NLP. Text mining jest do transformacja nieustrukturyzowanych danych tekstowych w uporządkowany format w celu otrzymania wartościowych informacji, które pozwalają na zrozumienie tekstu. Ponadto przy użyciu różnych narzędzi statystycznych czy algorytmów uczenia maszynowego jak maszyna wektorów nośnych czy deep learning można badać związek między danymi w tekście. Same dane mogą przyjmować format :

- Ustrukturyzowany (Dane są w ustandaryzowanej formie tabelarycznej, składającej się z wielu wersów i kolumn. W takiej formie łatwiej przechowuje się i procesuje dane, można dokonywać analizy czy też budować modele predykcyjne)
- Nieustrukturyzowany (Takie dane nie mają z góry zdefiniowanego formatu, może to być tekst z różnych źródeł jak media społecznościowe recenzje produktów czy formaty multimedialne takie jak pliki audio czy wideo)
- Pół-ustrukturyzowany (Jest to pewne połącznie danych ustrukturyzowanych i nieustrukturyzowanych, dane są w pewny sposób uporządkowany, ale nie spełniają kryteriów relacyjnej bazy danych. Przykładem takiego formatu są pliki XML, JSON czy też HTML)

Szacuję się, że ok. 80% danych świecie jest w formie nieustrukturyzowanej co czyni text mining wyjątkowo cennym narzędziem do analizy danych tekstowych.

Przetwarzanie języka naturalnego (ang. „Natural-language processing” – NLP) jest to zbiór technik pozwalających na zrozumienie komputerowi języka. W processingu tym wyróżnia się dwa etapy

- Preprocessing danych

- Rozwój algorytmu

Data preprocessing to zbiór metod służących do przygotowania oraz oczyszczenia danych dla komputera w celu uzyskania prostszego zbioru, nie zawierającego zbędnych elementów by zyskać na wydajności rozwiązania. Wśród metod tych można wyróżnić:

- Tokenizację
- Eliminowanie słów stop listą
- Stemming
- Dzielenie na części zdania

Metoda tokenizacji polega na dzieleniu tekstu na mniejsze części i zostanie bardziej opisana w dalszej części pracy. Stemming polega zaś na usuwaniu formantów ze słów w celu uproszczenia zbioru tekstowego, dzięki temu wyciągane jest samo zdanie słowa. Jako przykład można podać słowa „programming" i „programmer" (Kibble, 2013). Słowa ta co prawda nie mają takiego samego znaczenia, lecz wskazują na tą samą dziedzinę, więc można je uprościć za pomocą stemmingu. Wówczas dostajemy w rezultacie słowo „programm", które będzie częściej się powtarzać oraz z racji swojej krótszej nazwy, jest też bardziej oszczędne pamięciowo i obliczeniowo dla komputera. (Kibble, 2013)

Dzielenie na części zdania, polega natomiast na przypisaniu słowom w tekście odpowiednich poziomów będącymi takimi częściami zdania jak np. podmiot, orzeczenie czy dopełnienie.

2.2 Analiza Sentymentu oraz prawo Zipfa

Analiza sentymentu polega na wykorzystaniu przetworzenia języka naturalnego, analizy tekstu, lingwistyki komputerowej do systematycznego identyfikowania, wyodrębniania, określania emocjonalnego wydźwięku źródła tekstowego, czy był on pozytywny, negatywny bądź neutralny. Znaczenie analizy sentymentu jest tak duże ze względu na to że może ono być w pełni zautomatyzowane, dzięki czemu oszczędza to dużo pracy człowieka i jego czasu. Ponadto jest to narzędzie o rosnącej popularności, które używane jest w takich dziedzinach jak: e-commerce, marketing (badanie satysfakcji klientów z produktu), polityka (badanie nastrojów społecznych) czy też badanie rynku. Ogromną skarbnicą takich danych są różne platformy społecznościowe jak: twitter, facebook, reddit gdzie codziennie miliony użytkowników udziela wpisów na różne tematy (Silge i Robinson, 2017).

Przeprowadzanie analizy sentymentu należy zawsze zacząć od stworzenia lub pobrania leksykonu (czyli specjalnego zbioru danych) zawierający poszczególne wyrażenia i ich scoring sentymentu. W następnym kroku algorytm sam zaczyna po podziale tekstu na określone części (takie jak np. n-gramy, czy też poszczególne tweety; każdy tweet może mieć maksymalnie 140 znaków) klasyfikować wydźwięk emocjonalny danego fragmentu tekstu. Algorytm wówczas

dokonuje bilansu scoringu w oparciu o każde poszczególne słowo w analizowanym fragmencie i podaje wynik sentymentu. Zakres wyników może przyjmować różny charakter: od ciągłego gdzie wartości poniżej 0 odpowiadają negatywnemu wydźwięku zaś wyniki powyżej 0 wskazują na coraz bardziej pozytywny ton wypowiedzi. Wyniki też mogą być w postaci dyskretnej: wówczas wszystkie negatywne słowa są w kategorii negatywnej (lub -1), zaś pozytywne w kategorii pozytywnej (lub 1).

Do głównych wyzwań i trudności z jakimi analiza sentymentu się musi mierzyć są

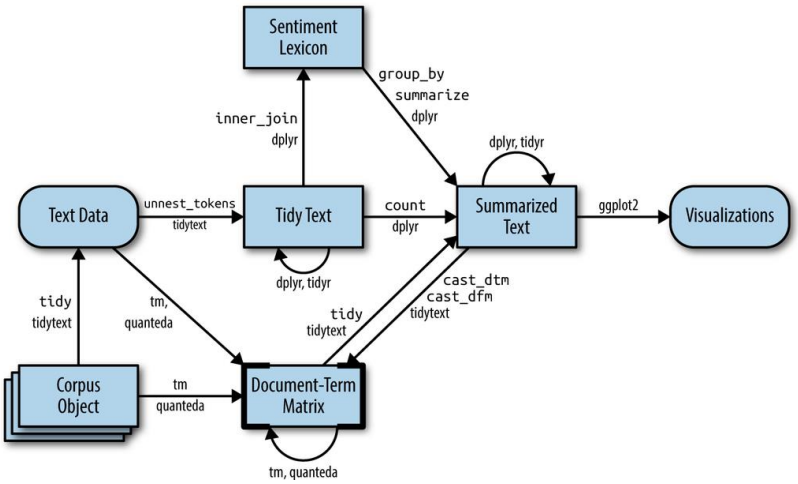
- Zaprzeczenia
- Złożone zdania o kontrastującym wydźwięku
- Sarkazm
- Rozpoznanie nazwanych podmiotów (Named-entity recognition)

Przykładem zaprzeczenia są wszelakie zdania z takim wyrazami jak „nie", „niezbyt" czy „wcale". Przypadkiem tego można uznać za dwa zdania, „Biorąc pod uwagę wszelkie wydarzenia, należy uznać dzisiejszy dzień za udany" wskazuje na pozytywny wydźwięk wypowiedzi z którym analiza sentymentu nie powinna mieć błędu. Dla zdania „Biorąc pod uwagę wszelkie wydarzenia, nie należy wcale uznać dzisiejszy dzień za udany" z kolei już nie można być pewnym poprawności wyniku analizy ze względu na to, że przy zsumowaniu lub uśrednieniu wyniku sentymentu poszczególnych słów wciąż wynik może wskazywać na pozytywny ton wypowiedzi mimo, że w rzeczywistości tak nie jest.

W przypadku zdań o złożonych o kontrastującym wydźwięku problemem z jakim algorytm musi się zmierzyć jest występowanie słów o kontrastującym wydźwięku co może wpływać na błędny wynik analizy. Przykładem takiego zdania może być : „Dzisiaj rano czułem się okropnie, ale mimo to na przyjęciu bawiłem się przednio". Takie zdanie jest dla modelu ciężkie do poprawnego rozpoznania ze względu na obecność słów o bardzo rozbieżnym wydźwięku, występują tu słowa o bardzo pozytywnym wydźwięku jak „bawiłem się" i „przednio", ale też jest słowo „okropnie", które ma bardzo negatywny wynik sentymentu.

Modele analizy sentymentu nie są natomiast w ogóle w stanie poprawnie sklasyfikować opinii sarkastycznych. Dzieje się tak ponieważ, wydźwięk emocjonalny nie jest przenoszony wprost w słowach dlatego też takie zdanie „Bardzo się cieszę, że mój pociąg przyjedzie z dwugodzinnym opóźnieniem" nie jest w stanie być prawidłowo sklasyfikowanym przez jakąkolwiek analizę sentymentu.

2.3 DocumentTermMatrix i TermDocumentMatrix



Wykres 1: Schemat działania metod biblioteki tidytext przy pracy z macierzami DTM i TDM Źródło: „Text mining with R" – Julia Silge i David Robinson

W text miningu do jednych z najczęściej używanych struktur analitycznych należą macierze DocumentTermMatrix i TermDocumentMatrix. Macierz DocumentTermMatrix jest to macierz w której:

- Każdy wiersz reprezentuje dokument (czyli źródło tekstowe takie jak książka czy artykuł)
- Każda kolumna reprezentuje term (czyli słowo)
- Każda wartość reprezentuje ilość danego termu w danym dokumencie

Ponieważ taka macierz przeważnie składa się z zer macierze DTM są głównie implementowane w postaci rzadkiej (ang. „sparse matrix"). Macierze rzadkie są często używaną strukturą w obliczeniach i informatyce ze względu na łatwość skompresowania i co za tym idzie, mniejszym obciążeniem pamięciowym. Mierzenie rzadkości (ang. „sparsity") takiej macierzy jest dane wzorem:

$$Rzadkość = \frac{Ilość\ zer\ w\ macierzy}{Ilość\ wszystkich\ wartości\ w\ macierzy} \times 100\%$$

Transponowana macierz DTM jest nazywaną macierzą TermDocumentMatrix. Wtedy w to w danej kolumnie znajdują się liczności poszczególnych termów w danym dokumencie (Kibble, 2013). Wówczas w bardziej wygodny sposób można wyliczać takie metryki jak częstotliwość występowania termu tf (term-frequency), idf (inverse document frequency) czy też współczynnik tf-df.

Prawo Zipfa głosi, że częstotliwość występowania danego termu w dokumencie jest odwrotnie proporcjonalna do jego rangi istotności, co oznacza, że iloczyn rangi istotności termu i częstotliwości występowania jest stała (Silge i Robinson, 2017).

2.4 N-gramy

Jedną z technik działania na ogromnych i skomplikowanych zbiorów tekstowych jest dzielenie tekstu na tzw. n-gramy. N-gramy są to sekwencje składające się z n słów. Pozwala to na łatwiejsze operowanie na tekście i ograniczenie czasowe działania algorytmów. Do uproszczenia tekstu używa się także stoplisty. Stoplista jest to zbiór słów, które nie mają same w sobie znaczenia, więc nie są potrzebne w analizie znaczenia tekstu (Silge i Robinson, 2017). Są to różnego rodzaju spójniki i przyimki, które służą do logicznego sensu wypowiedzi, lecz nie nadają jej żadnego znaczenia, tonu czy wydźwięku. Ilość n-gramów w jednym zdaniu lub fragmencie tekstu K można określić wzorem:

$$Ngramy_K = X - (N - 1)$$

Gdzie X oznacza ilość słów w zdaniu K

W modelach i analizie NLP najczęściej używa się bigramów (sekwencji dwuwyrazowych) oraz trigramów (sekwencji trójwyrazowych). N-gramy są używane m.in. do budowania języków, gdzie sprawdzana jest poprawność pisowni czy też skracania źródeł tekstowych w celu pozbycia się niepotrzebnych wyrazów. Za przykład można podać to zdanie: „Dzisiaj przewidywane są przelotne opady”. Gdy rozbije się je na bigramy otrzyma się wówczas:

- „Dzisiaj przewidywane”
- „przewidywane są”
- „są przelotne”
- „przelotne opady

Zgodnie ze wzorem otrzymano ze zdania składającego się z 5 słów, 4 bigramy. Z kolei gdy to zdanie rozbije się na trigramy to powstaną poszczególne człony:

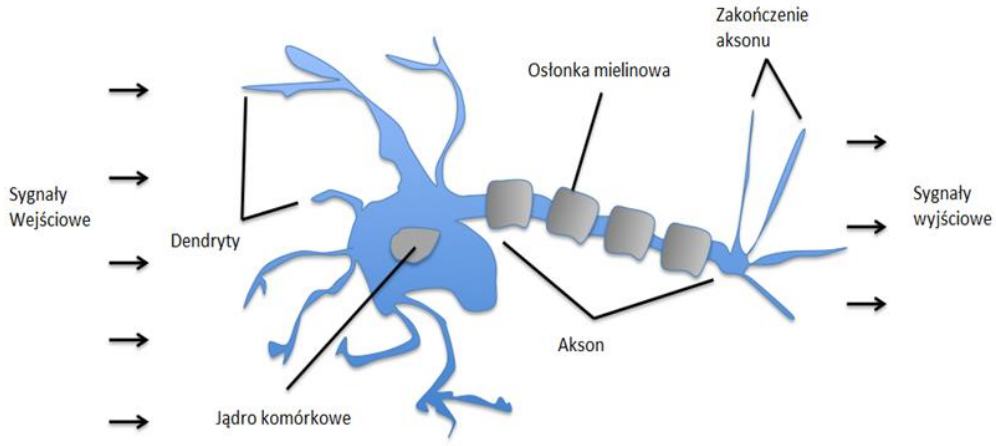
- „Dzisiaj przewidywane są”
- „przewidywane są przelotne”

- „są przelotne opady”

Zgodnie ze wzorem otrzymano ze zdania składającego się z 5 słów, 3 trigramy.

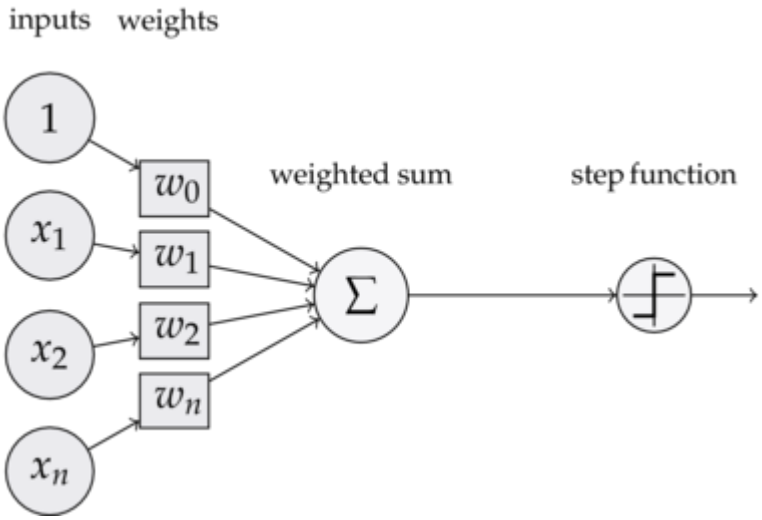
2.5 Czym są sieci neuronowe i jakie są ich zastosowania

Do klasyfikacji komentarzy użyto metody uczenia głębokiego (ang. „deep learning”). Opiera się ona o struktury matematyczne zwane sieciami neuronowymi. Nazwa tych struktur, nie jest przypadkowa, ich działanie jest inspirowane naturą, konkretnie mózgami zwierząt. Sam neuron jest komórką składająca się z ciała komórkowego (periakrionu), czyli wypustek określanych mianem dendrytów i aksonów. Neurony potrafią odbierać i przesyłać sygnały elektryczne oraz wszelakie informacje. Dendryt z kolei to wypustka stanowiąca przedłużenie komórki nerwowej i odpowiada za odbieranie impulsów i przesyłanie ich do ciała komórki w celu ich integracji. Dendryt to największa część neuronów – stanowi do 90% powierzchni wielu z nich. Akson z kolei służy do przesyłania informacji z ciała komórkowego do reszty komórek nerwowych. Proces ten znany jest jako neurotransmisja.



Wykres 2:Schemat budowy neuronu
Źródło: www.healthline.com

W 1958 roku Frank Rosenblatt opracował i zbudował najprostszy model sieci neuronowej zwanej perceptronem. W swojej najprostszej wersji perceptron był zbudowany z dwóch warstw neuronów reprezentujących odpowiednio wejście i wyjście. Rosenblatt odkrył też ważną właściwość perceptronu, którą przedstawił swoim twierdzeniu: „Jeżeli tylko istnieje taki wektor wag *w*, przy pomocy którego element perceptronowy odwzorowuje w sposób zbiór oczekiwanych wartości wyjściowych, to istnieje metoda uczenia tego elementu gwarantująca zbieżność do wektora w.

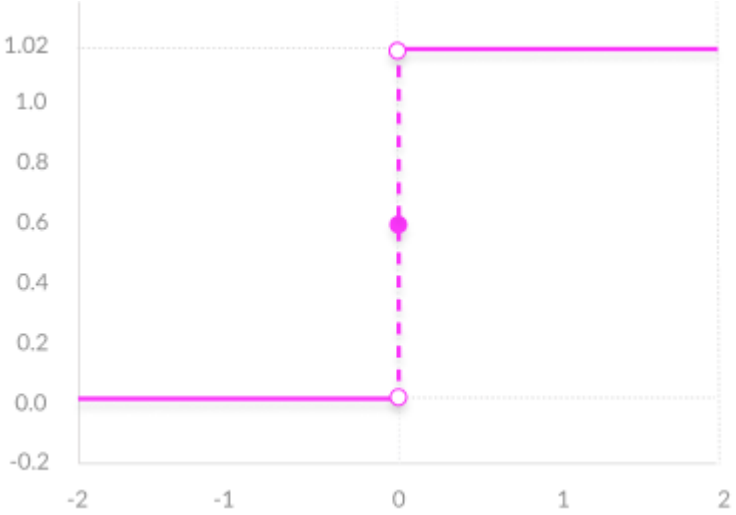


Wykres 3: Schemat perceptronu Rosenblatta
Źródło: www.statworx.com

Na wykresie 3 przedstawiono perceptron z tzw. uprzedzeniem (ang. „bias"), czyli do parametrów wejściowych dodano również wartość równa 1 razem ze swoją wagą w celu łatwiejszego przypisania. Obliczenie sumy ważonej jest dane wzorem:

$$\sum_{i}^{n} w_i \times x_i$$

Gdzie x oznacza i-ty parametr wejściowy a w jego wagę. W kolejnym kroku suma ważona trafia to funkcji aktywacji, gdzie wartość sumy jest przekształcona na wyjście. Pierwotnie były to binarne funkcje dyskretne, które w zależności od tego czy suma jest większa od wartości progowej zwracały 1 lub 0 , bądź 1 lub – 1.



Wykres 4: Przykład binarnej funkcji aktywacji
Źródło: www.mygreatlearning.com

Taka bardzo prosta funkcja aktywacji ma jednak dwie duże wady. Z racji tego, że jest binarna oznacza to, że nie nadaje się do rozwiązania problemów klasyfikacji, gdzie ma się do czynienia z więcej niż 2 klasami rozpoznania. Kolejnym jeszcze większym problemem jest fakt, że gradient takiej funkcji wynosi 0 co poważnie utrudnia przeprowadzenie korekcji wag algorytmem propagacji wstecznej.

Rozwiązaniem tego problemu stanowią nieliniowe funkcje aktywacji:

- Pozwalają przeprowadzenie algorytmu propagacji wstecznej ze względu na powiązanie pochodnej funkcji aktywacji z wejściem co pozwala na lepsze zrozumienie, które wagi w neuronie są w stanie dokonać lepszej predykcji
- Pozwalają na składanie wielu warstw neuronów co pozwala na uzyskanie wyjścia w postaci nieliniowej kombinacji parametrów wejściowych przepuszczonych przez wiele warstw.

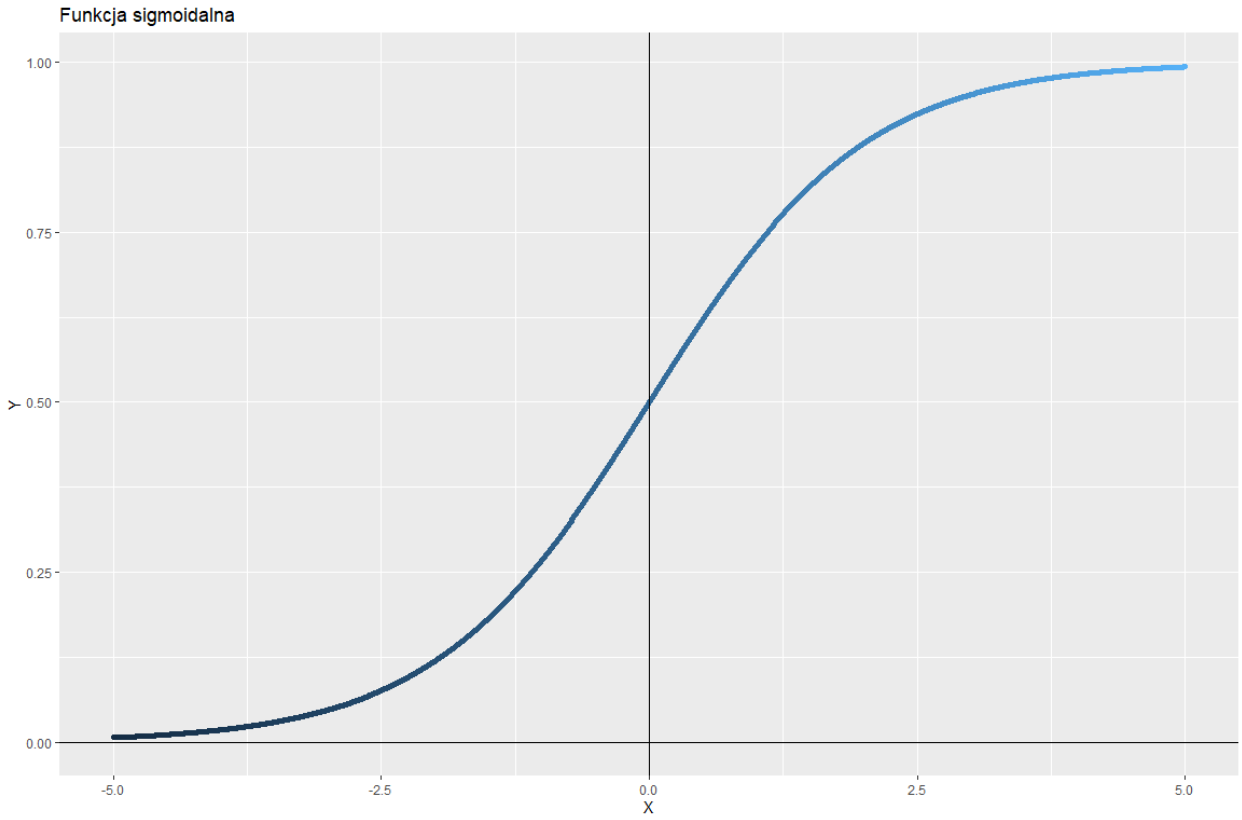
Do najczęściej używanych funkcji aktywacji należą:

1. Sigmoidalna funkcja aktywacji

Funkcja ta jest dana wzorem:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Jest to funkcja bardzo często używana do modelowania prawdopodobieństwa ze względu na zakres jej wartości równy <0, 1>. Do kolejnej zalety tej funkcji należy jej różniczkowalność, która zapewnia łagodny gradient przez co nie ma skoków w wartościach wyjściowych.



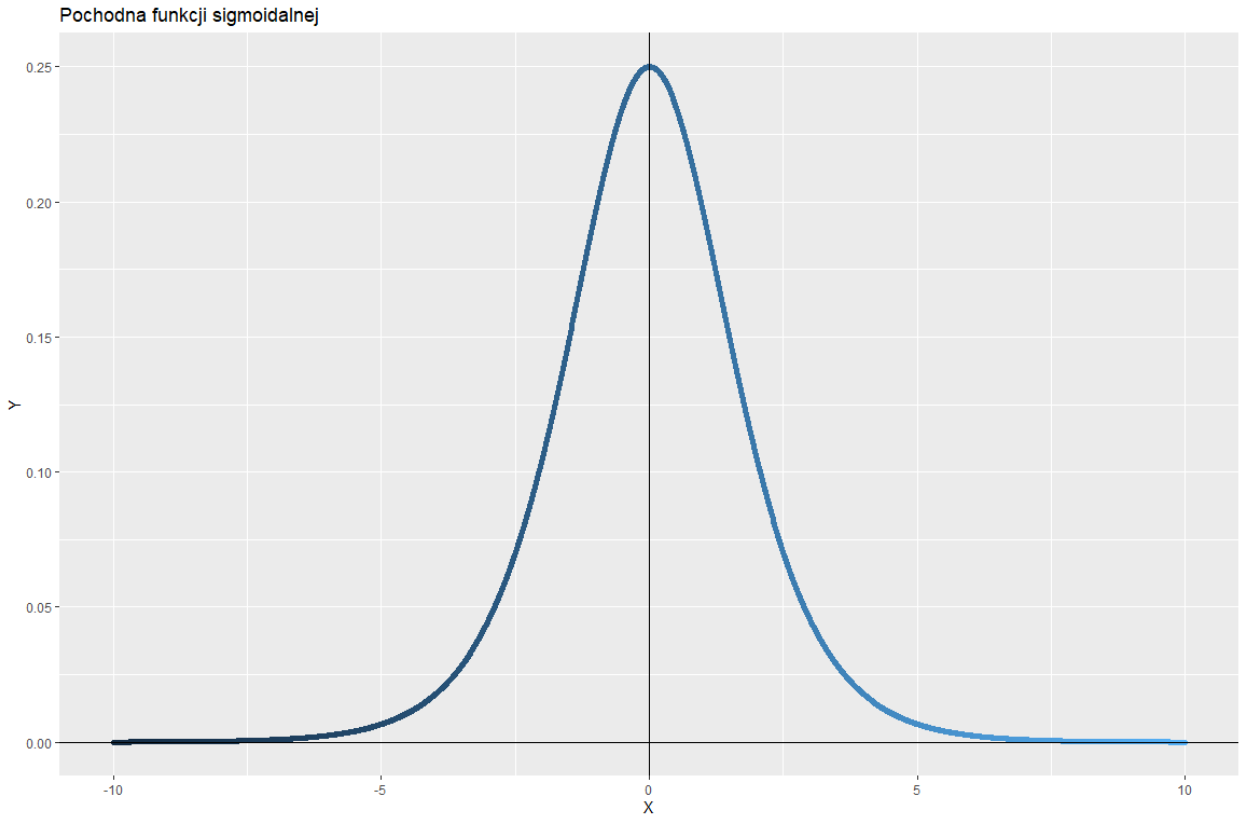
Wykres 5: Przedstawienie funkcji sigmoidalnej

Źródło: Opracowanie własne

Wadą funkcji sigmoidalnej jest natomiast mały zakres wartości jej pochodnej. Pochodna jest ta dana wzorem:

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

Gdzie $\sigma(x)$ oznacza funkcję sigmoidalną.



Wykres 6: Przedstawienie pochodnej funkcji sigmoidalnej

Źródło: Opracowanie własne

Wykres 6 pokazuje, że wartości są znaczące tylko na przedziale argumentów <-5,5>. Dla pozostałych argumentów, krzywa pochodnej funkcji sigmoidalnej zbiega do 0. To oznacza, że wartość gradientu w tych argumentach będzie bardzo niska. Jest to negatywne zjawisko, gdyż w przypadku gdy wartość gradientu osiąga 0, sięc przestaje się uczyć i dotyka ją tzw. problem zanikającego gradientu (ang. „vanishing gradient problem"). Kolejnym problemem jest kwestia wartości wyjściowych (Baheti, 2022). Na wykresie 5 można zauważyć, że krzywa nie jest symetryczna względem zera co oznacza, że wyjścia u wszystkich neuronów będzie tego samego znaku. Fakt ten utrudnia proces trenowania sieci.

2. Funkcja ReLU

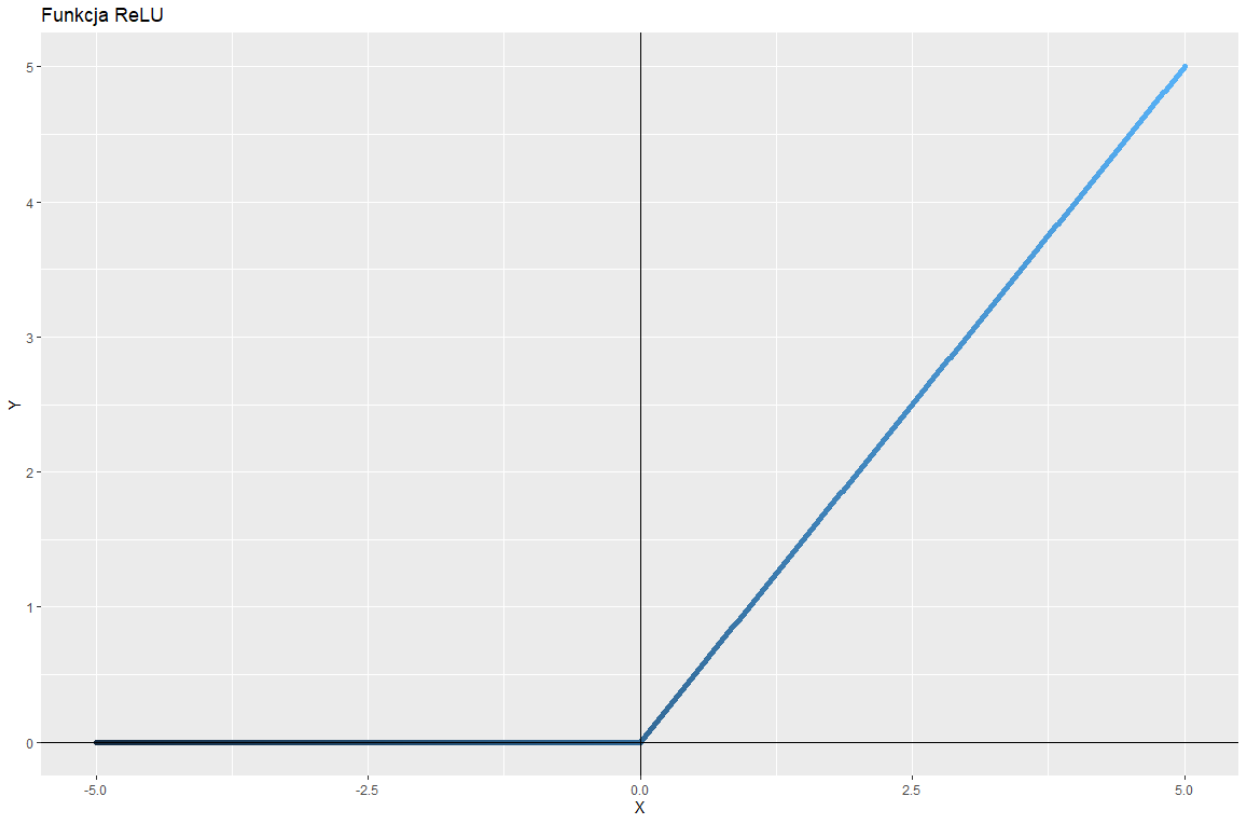
Funkcja ta jest dana wzorem:

$$f(x) = \max(x, 0)$$

Akronim ReLU oznacza „Rectified Linear Unit". Skorygowana funkcja aktywacji ReLU jest użyteczna ze względu jej różniczkowalność oraz możliwość korygowania wag algorytmem propagacji wstecznej przy jednoczesnym zapewnieniu wydajności obliczeniowej. Cechą

charakterystyczną funkcji ReLU jest to, że nie wszystkie neurony są aktywowane w tym samym czasie. W przypadku gdy wynik transformacji ReLU będzie równy 0, neuron zostanie deaktywowany. Do zalet funkcji ReLU należą:

- Większa wydajność obliczeniowa w stosunku do funkcji sigmoidalnej czy tanh ze względu na fakt, że tylko niektóre neurony są aktywowane
- Funkcja ReLU przyspiesza zbieganie spadku gradientu do globalnego minimum funkcji straty

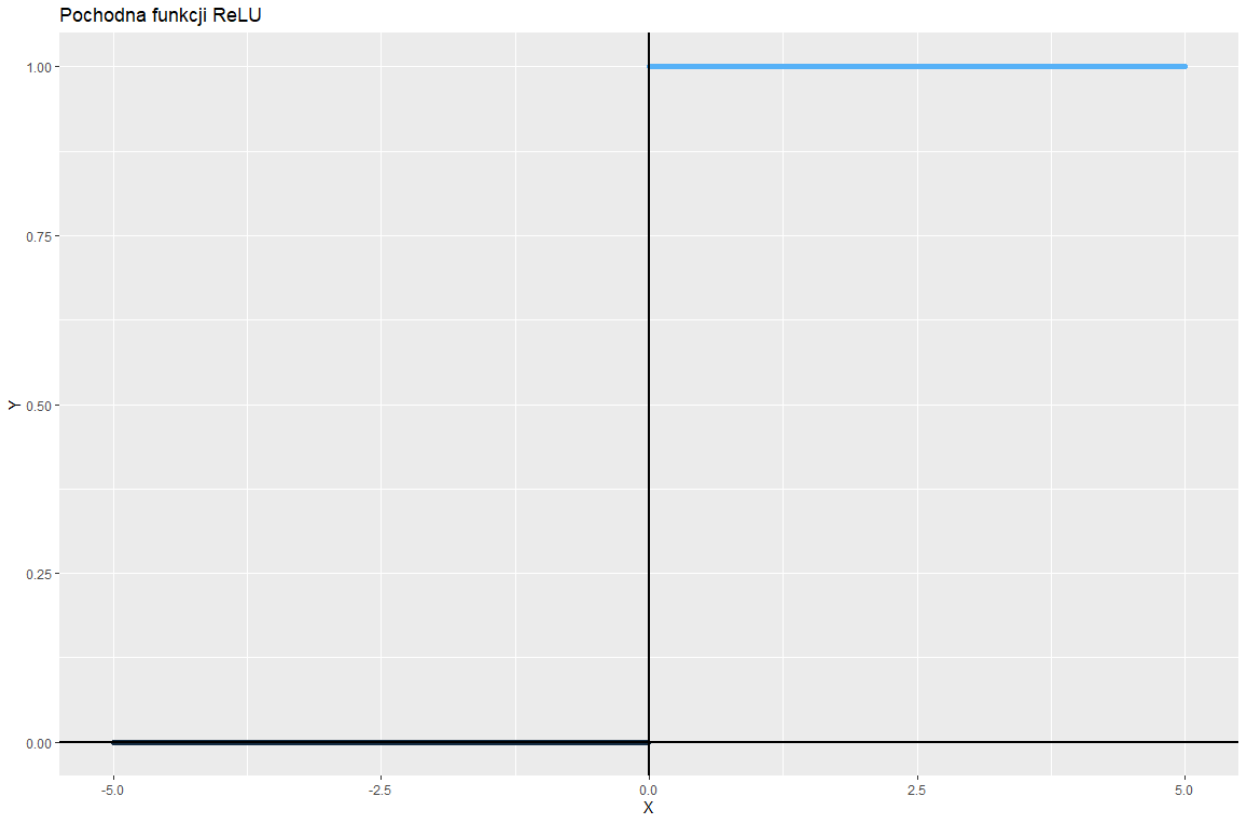


Wykres 7: Przedstawienie funkcji reLU
Źródło: Opracowanie własne

Do wad funkcji ReLU należy natomiast tzw. problem umierającego ReLU (ang. „dying ReLU problem). Pochodną tą można opisać wzorem:

$$f'(x) = g(x) = 1 \; jeżeli \; x \geq 0$$

$$g(x) = 0 \; jeżeli \; x < 0$$



Wykres 8: Przedstawienie pochodnej funkcji ReLU

Źródło: Opracowanie własne

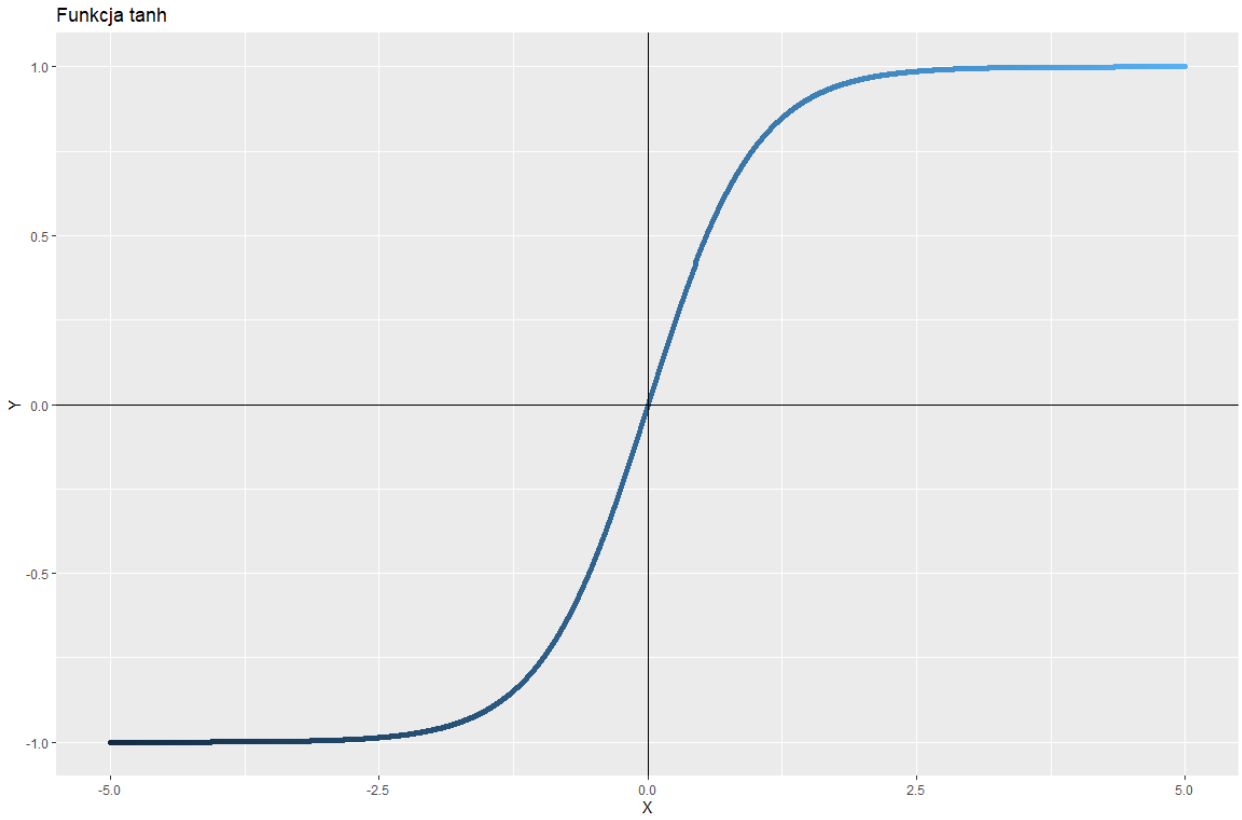
Na wykresie 8 widać, że dla ujemnych argumentów wartość gradientu będzie wynosić 0. Z tego powodu, dla niektórych, podczas kroku propagacji wstecznej, wagi w niektórych neuronach nie zostaną poprawione. To z kolei może skutkować w utworzeniu martwych neuronów, które nigdy nie zostaną aktywowane.

3. Funkcja tanh

Funkcja ta jest dana wzorem:

$$f(x) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})}$$

Funkcja tanh jest bardzo podobna do funkcji sigmoidalnej. Jej krzywa podobnie jak krzywa sigmoidalna układa się w kształt przypominający literę „S". Różni się ona natomiast zakresem wartości, funkcja tanh przyjmuje wartości także ujemne czyli od -1 do 1. Funkcja ta zbiega dla coraz mniejszych argumentów do -1, z kolei dla coraz większych argumentów wartość funkcji zbiega do 1 (Baheti, 2022).

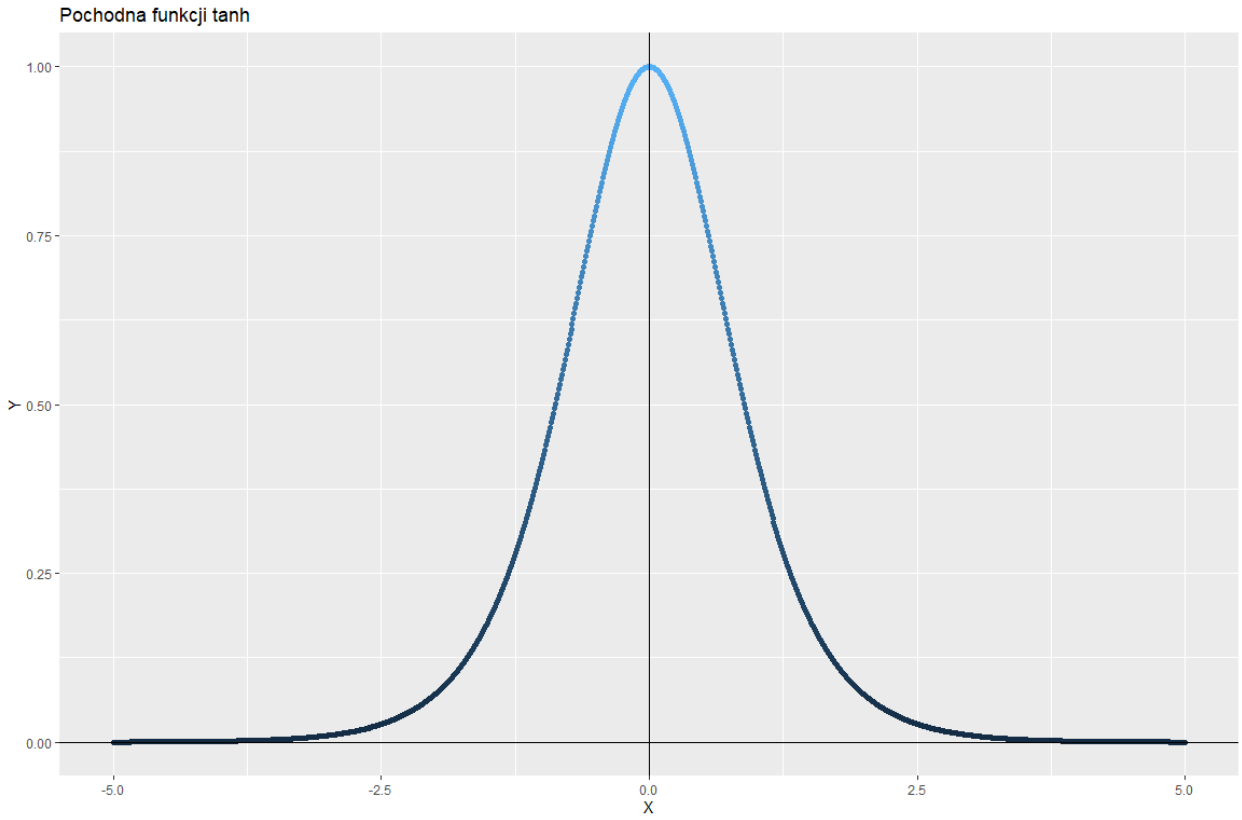


Wykres 9: Przedstawienie funkcji tanh

Źródło: Opracowanie własne

Funkcja tanh ma jednak bardzo podobne ograniczenia co funkcja sigmoidalna. Podobnie tylko dla wąskiego zakresu argumentów gradient ma wartości znaczniej różniące się od 0. Pochodna funkcji tanh jest dana wzorem:

$$tanh'(x) = 1 - tanh^2(x)$$



Wykres 10: Przedstawienie pochodnej funkcji tanh

Źródło: Opracowanie własne

Na wykresie 10 można zauważyć, że również problem zanikającego gradientu dotyczy również funkcję tanh. Różnicą natomiast między krzywą funkcji sigmoidalnej a tanh jest większa stromość krzywej funkcji tanh.

4. Funkcja softmax

Jest to funkcja, która za argument przyjmuje wektor prawdopodobieństw (mogą to też być wyliczone funkcją sigmoidalną predykcje). Aby ta funkcja zadziałała, musi zostać spełniony warunek: suma wartości w wektorze musi być równa 1. Następnie każdemu elementowi wektoru jest przypisywana wartość według następującego wzoru:

$$softmax(x_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

Dlatego dla wektora wyjść neuronów [0,34. 1.34. 0.85] funkcja softmax wyliczy wektor prawdopodobieństw [0,19; 0,5 i 0.31]. W następnej kolejności funkcja ta zwróci wektor zerojedynkowy, w którym najwyższej wartości wyjściowej zostanie przypisana jedynka, a reszcie zera. W rezultacie otrzymany zostanie końcowy wektor [0, 1, 0]. Funkcja softmax jest najczęściej używana w ostatniej warstwie sieci neuronowych w przypadkach klasyfikacji wielopoziomowych (Baheti, 2022).

Modele sieci neuronowych podobnie jak inne modele predykcyjne mogą być nadzorowane i trenowane. Taka optymalizacja sieci odbywa się poprzez korekcję wartości wag w neuronach. Do najpopularniejszego algorytmu korekcji wag należy tzw. algorytm propagacji wstecznej. Jest to algorytm używany w sieciach wielowarstwowych czyli takich sieciach neuronowych, które oprócz warstwy wejściowej i wyjściowej mają jeszcze warstwy ukryte. Algorytm ten opiera się na minimalizacji sumy kwadratów błędu uczenia z wykorzystaniem optymalizacji metody największego spadku (Korbicz, Obuchowicz i Uciński, 1994). Należy najpierw rozważyć błąd sieci ξ:

$$\xi = \sum_{\mu=1}^{P} \xi = \frac{1}{2} \sum_{\mu=1}^{P} \sum_{j=1}^{n} (y_j^{z\mu} - \varphi_j^{\mu})^2$$

Gdzie

$$\xi_\mu = \frac{1}{2} \sum_{j=1}^{n} (\delta_j^{\mu})^2$$

Przy czym n oznacz liczbę elementów w warstwie wyjściowej.

Problemem uczeni sieci neuronowych jest znalezienie globalnego minimum funkcji błędu ξ. Bardzo często używaną do tego metodą jest gradientowa metoda największego spadku. Polega to na iteracyjnym poszukiwaniu kolejnego lepszego punktu w kierunku przeciwnym do gradientu funkcji celu w danym punkcie. Stosując tę metodę, zmiana wagi połączenia w_{ji} powinna spełniać relację:

$$\Delta w_{ji} = -\eta \frac{\vartheta \xi}{\vartheta w_{ji}} = -\eta \sum_{\mu=1}^{P} \frac{\vartheta \xi_\mu}{\vartheta w_{ji}} = -\eta \sum_{\mu=1}^{P} \frac{\vartheta \xi_\mu}{\vartheta y_j^{\mu}} \frac{\vartheta y_j^{\mu}}{\vartheta w_{ji}}$$

Gdzie η oznacza współczynnik proporcjonalności.

W przypadku liniowych elementów przetwarzających zachodzi równości:

$$\frac{\vartheta \xi_\mu}{\vartheta \xi_j^{\mu}} = -\left(y_j^{z\mu} - y_j^{\mu}\right) = -\delta_j^{\mu}$$

$$\frac{\vartheta y_j^{\mu}}{\vartheta w_{ji}} = \frac{\vartheta \varphi_j^{\mu}}{\vartheta w_{ji}} = u_i^{\mu}$$

Stąd zostanie otrzymana korekta wagi

$$\Delta w_{ji} = \eta \sum_{\mu=1}^{P} \delta_j^{\mu} u_i^{\mu}$$

Po czym ostatecznie otrzymana jest tzw. „regułą delty", która jest dana wzorem:

$$w_{ji}^{n} = w_{ji}^{s} + \Delta w_{ji}$$

Gdzie górny indeks n oznacza nową, a s starą wagę. Reguła delty przy dostatecznie małym współczynniku proporcjonalności uczenia η, poszukuje zbioru wag minimalizującego funkcję błędu sieci liniowej. Należy pamiętać, że metoda propagacji wstecznej działa tylko dla wielowarstwowych sieci jednokierunkowych, nie zadziała ona na przykład w modelach generatywnych które opierają się o sieci rekurencyjne czyli takie sieci, które posiadają sprzężenie zwrotne (Korbicz, Obuchowicz i Uciński, 1994).

Wśród warstw ukrytych możemy rozróżnić wiele ich rodzajów: są to na przykład sieci gęste czyli sieci składające się od kilkunastu do kilkudziesięciu neuronów (może być w jednej warstwie 16, 32 czy nawet 64 neuronów). Najczęściej używaną funkcją aktywacji w tego rodzaju warstwie jest funkcja ReLU. Warstwę tę też można użyć jako warstwę końcową (np. w postaci neuronów z funkcją sigmoidalną). Innymi, szeroko stosowanymi warstwami sią sieci konwolucyjne. W sieciach tych używa się filtra nazywanego jądrem (ang. „kernel"). Jednym z hiperparametrów sieci konwolucyjnej jest rozmiar jądra. Mniejszy rozmiar jądra wpływa na lepszą dokładność w uzyskaniu informacji kluczowych z danych wejściowych. Wpływa to również na mniejszego zmniejszenia dalszych warstw co w rezultacie daje głębszą architekturę (tj. więcej warstw w sieci neuronowej). Większy rozmiar z kolei kosztem odbija się na mniejszej dokładności lecz lepszej generalizacji problemu, gdy np. w danych wejściowych nie trzeba przykładać uwagi do szczegółów. Podczas uczenia sieci neuronowej dokonywana jest korekta wag w filtrze. Następnie uogólniony wynik jest poddawany funkcji aktywacji – najczęściej używana jest wówczas funkcja ReLU. Sieci CNN (ang. „Convolutional Neural Networks") są szeroko używane w rozpoznawaniu i klasyfikacji obrazów (filtry 2D) czy też w analizie tekstowej (filtr 1D). Po przejściu wartości przez filtr wyniki przechodzą do warstwy poolingu. Pooling służy do upraszczania obrazu bądź tekstu co wpływa pozytywnie na wydajność modelu ze względu na mniejszą ilość parametrów do przetworzenia. Wyróżnia się trzy rodzaje poolingów:

- Max pooling (z danych elementów wybiera się ten o największej wartości)
- Min pooling (z danych elementów wybiera się ten o najmniejszej wartości)
- Average pooling (Wówczas wynik się uśrednia)

Jako przykład można podać macierz pikseli :

$$\begin{bmatrix} 129 & 243 \\ 85 & 100 \end{bmatrix}$$

W max poolingu jako wynik otrzymamy piksel o wartości 243, w min poolingu natomiast 85. Z kolei w average poolingu wynikiem będzie średnia wartość czyli 139. Max pooling jest ogółem stosowany przy analizie sentymentu ze względu na to że lepiej się w tekście skupić na wyrazach o mocnym wydźwięku. Do zapobiegania nadmiernego przeuczania się sieci używa się warstw dropoutu. Polega to na losowym wyłączaniu niektórych połączeń między neuronami. Powoduje to, że sieć nie łapie zbyt szybko wyuczonego wzorcu przez co problem overfittingu jest znacznie zredukowany.

Oprócz rozpoznawania i klasyfikacji obrazów czy tekstów, sieci neuronowe można wykorzystywać również do analizy video czy też zwykłych danych tabelarycznych. Ponadto można też wyróżniamy sieci generatywne, są to specjalne sieci rekurencyjne (czyli takie, które w odróżnieniu do jednokierunkowych sieci neuronowych, posiadają sprzężenie zwrotne w swojej topologii), które potrafią generować dane, a nawet takie obiekty jak zdjęcia, dźwięk czy tekst. Sieci generatywne można np. zastosować w odtworzeniu bardzo starych nagrań czy przy odnowieniu jakichś starych zdjęć z brakującymi elementami.

3. Analiza tekstu oraz weryfikacja komentarzy w języku R

3.1 Opis zbioru danych

Zbiór danych dotyczy komentarzy pacjentów na użyty przez nich w określonej dolegliwości. Komentarze były zbierane z różnych stron poświęconych recenzjom leków poprzez współpracę uniwersytetu Kansas State i Uniwersytetu Technicznego w Dreźnie. Wszystkie komentarze zostały napisane w języku angielskim, lecz nie podano informacji o krajach z których pacjenci pochodzili. Zbiór danych liczy łącznie ponad 215 tysięcy obserwacji. Oceniono w nim 3436 leków na 885 różnych dolegliwości. Dane pochodzą z lat 2008-2017 i składają się z poszczególnych zmiennych:

- drugName: nazwa leku
- condition: dolegliwość lub choroba
- review: treść komentarza
- rating: ocena, którą wystawił pacjent w skali 1-10
- date: data opublikowania komentarza
- usefulCount: ilość użytkowników, którzy uznali dany komentarz za przydatny

Zbiór danych jest dostępny pod linkiem (Kallumadi i Gräßer, 2018):

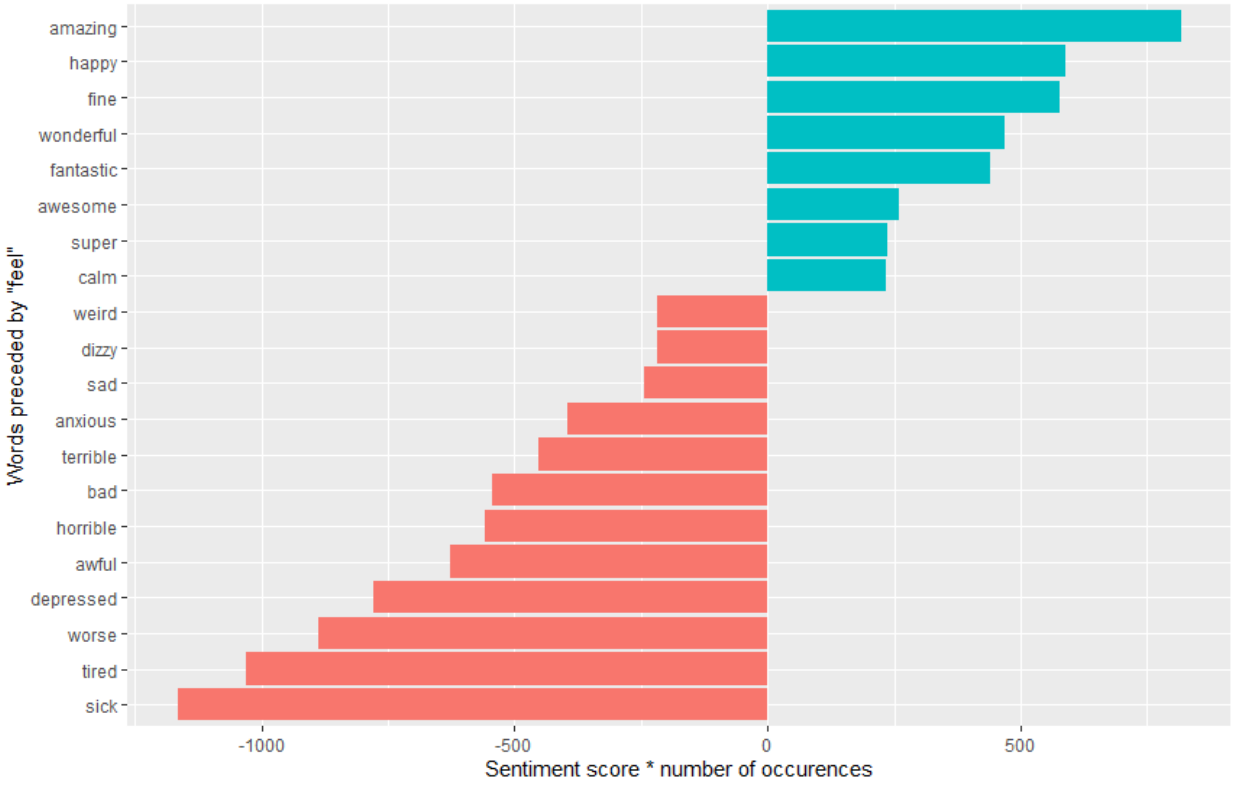
3.2 Analiza sentymentu oraz podziału na n-gramy przy pomocy pakietu tidytext

Do analizy sentymentu użyto biblioteki tidytext, aby pozbyć się niepotrzebnych słów tzw. „stop words". Są to słowa o małym znaczeniu takie jak np. spójniki („ponieważ", „oraz", „bo") czy też słowa popularne („mp3", „pc"). Słowa te nie wpływają na identyfikację tekstu dlatego też usuwa się je w celu zredukowania wielkości zbiorów i oszczędności pamięci operacyjnej. Następnie przy pomocy biblioteki tidytext pobrany został leksykon AFINN (Technical University of Denmark, 2011), który jest słownikiem w którym każde słowo ma przypisaną wartość sentymentu. W tym leksykonie słowa o negatywnym wydźwięku przyjmują ujemne wartości, zaś te o pozytywnym zwracają wartości dodatnie (np. wyrazy „amazing" czy „breathtaking" zwracają 5, natomiast wulgaryzmy przyjmują wartość -5) (Silge i Robinson, 2017). W dalszej kolejności obliczane są liczności n-gramów czyli występujących obok siebie *n* słów. Dla tego przypadku za *n* przyjęto 2 gdyż można w ten sposób ukazać związek między bezpośrednio sąsiadującymi ze sobą słowami oraz obliczyć ich kontrybucję na podstawie częstości ich występowania oraz wartości sentymentu. Po obliczeniu wystąpienia słów obliczona została kontrybucja każdego słowa wzorem (Silge i Robinson, 2017):

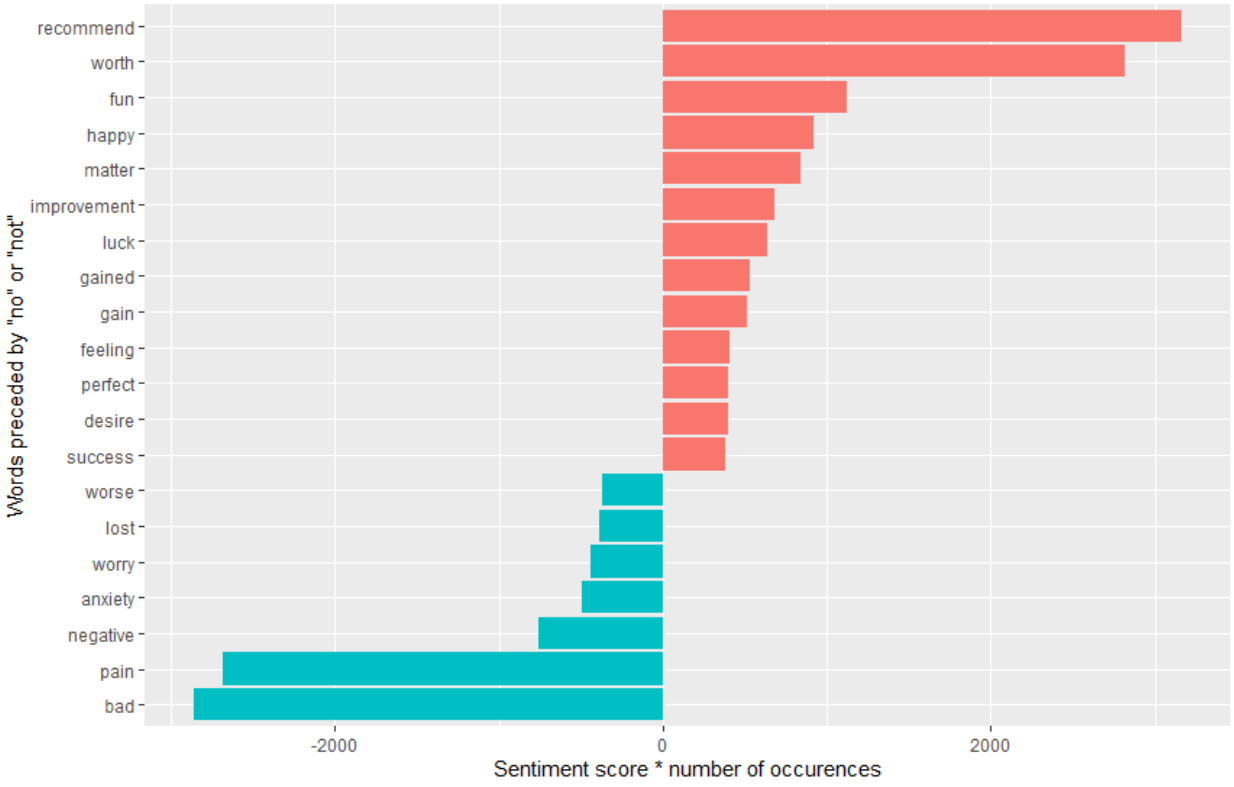
$$Kontrybucja = Wartość\ sentymentu\ słowa * ilość\ wystąpień$$

Gdzie wartość sentymentu słowa oznacza wartość sentymentu dla danego słowa w leksykonie AFINN, a ilość wystąpień dotyczy ilości wystąpień w dokumencie (dla tego przypadku wzięto pod uwagę cały zbiór komentarzy).

Kontrybucja pozwala na obliczenie jak bardzo dany term (czyli słowo) lub n-gram wpływa na wydźwięk tekstu (czy jest to zdanie nacechowane pozytywnie – przy wypadkowej kontrybucji o wartości większej niż 0 lub negatywnie w przypadku gdy łączna kontrybucja danego zdania bądź tekstu jest ujemna). W następnym kroku zbadano odczucia pacjentów wynikające z komentarzy.



Wykres 11: Słowa które najczęściej były poprzedzone słowem „feel" w różnej odmianie
Źródło: Opracowanie własne



Wykres 12: Słowa które najczęściej były poprzedzone słowem "no" lub "not" o najwyższym scoringu bezwzględnym
Źródło: Opracowanie własne

Wykres 1 pokazuje, że z takimi słowami jak „feel” czyli „czuć” w kontekście pozytywnym są związane takie słowa jak „amazing”(niesamowicie), „happy”(szczęśliwy) czy „fine”(dobrze). Jeśli chodzi natomiast o słowa o negatywnym wydźwięku, które w dużym stopniu występowały ze słowem „feel” dominują słowa typowe dla chorób takie jak „sick”(chory), „tired” (zmęczony) czy „depressed”. Ważnym słowem jest również „worse”(„gorzej”, które sugeruje, że opinia pacjenta jest negatywna (Silge i Robinson, 2017)).

Wykres 2 pokazuje zaś, że z takimi słowami jak „not” lub „no” powiązane są takie słowa jak „recommend”(polecać), „worth”(warto), „fun”(zabawnie). W przypadku słów o negatywnym znaczeniu najczęściej występuje słowo „bad”(źle) oraz „pain”(ból). Kolory na wykresie 2 zostały specjalnie odwrócone ze względu na zmieniony kontekst, tutaj słowa o pozytywnym znaczeniu będą oznaczać, że opinia o leku była najprawdopodobniej negatywna, zaś połączenie słów „no” i „pain” czy „not” i „worry” będą wskazywać, że pacjent takowy lek poleca.

Wykres 13: Chmura słów najczęściej przedstawiających się w komentarzach
Źródło: Opracowanie własne

Wykres 3 przedstawia chmurę słów w tekście. Na zielono są słowa o pozytywnym

znaczeniu, na czerwono słowa o znaczeniu negatywnym. Słowa takie jak „pain", „anxiety", „depression" czy „symptoms" wskazują na problemy zdrowotne jakie mieli pacjenci. Z kolei słowa „helped", „recommend", „worth", „effective" skupiają się już bardziej na ocenie leku.

.W następnym kroku zbadano współczynnik tf-idf do zbadania wagi słów w oparciu o liczbę ich wystąpień dla poszczególnych leków. Współczynnik tf-idf jest obliczany wzorem:

$$tf - idf_{i,j} = tf_{i,j} \times idf_i$$

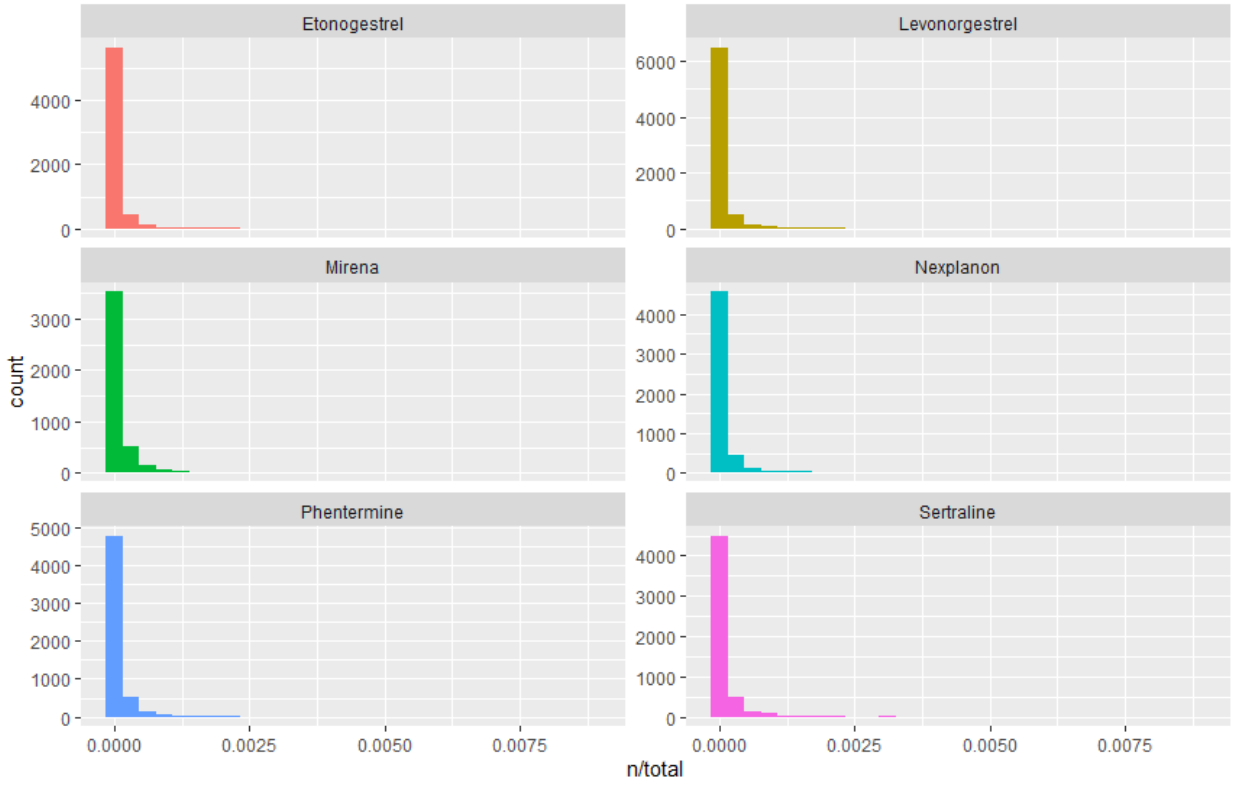
Gdzie:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

$$idf_i = \log \frac{|D|}{|\{d : t_i \in d\}|}$$

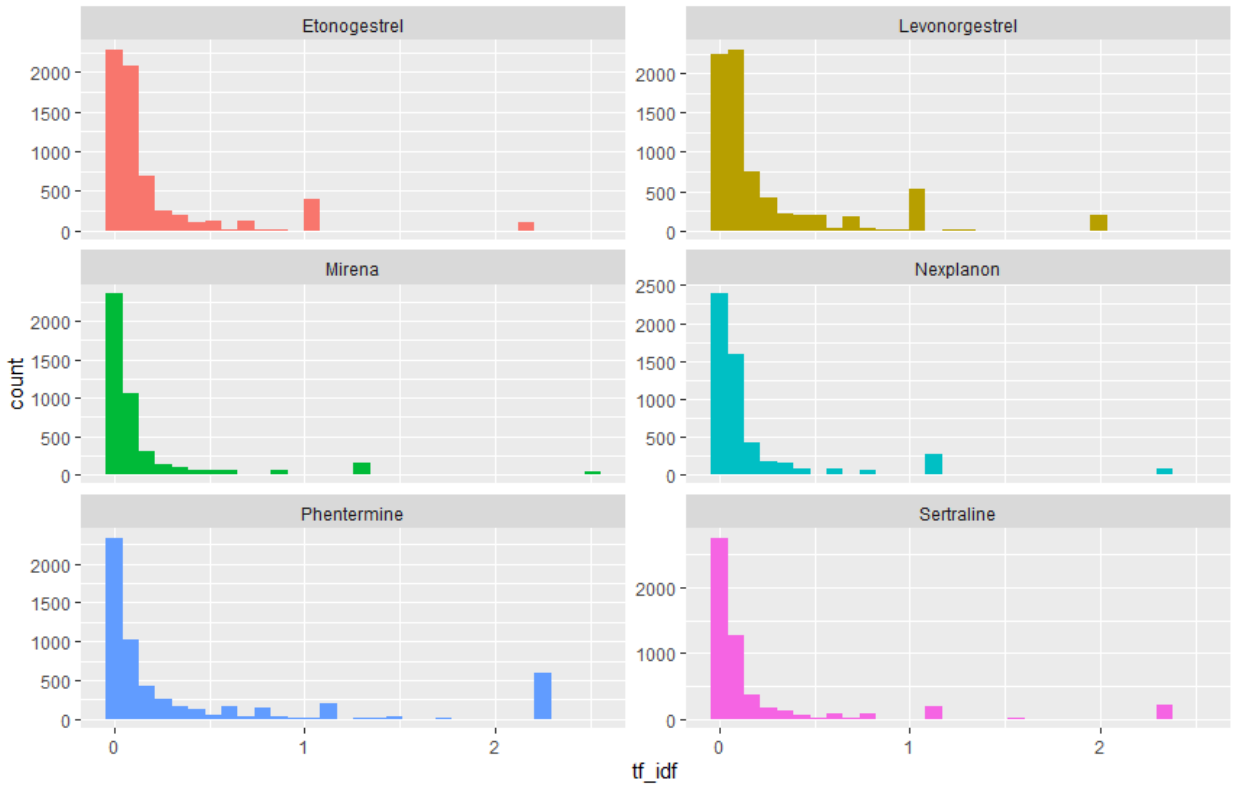
$|D|$ - oznacza liczbę dokumentów w korpusie, a $|\{d: t_i \in d\}$ oznacza liczbę dokumentów zawierających przynajmniej jedno wystąpienie i-tego termu (Silge i Robinson, 2017).

Współczynnik tf-idf jest istotny gdyż informuje czy zbiór danych tekstowych zawiera istotne informacje i można na nim robić analizę NLP w celu osiągnięcia określonego celu biznesowego. Analiza ta została przeprowadzona dla całego ogółu zbioru komentarzy.



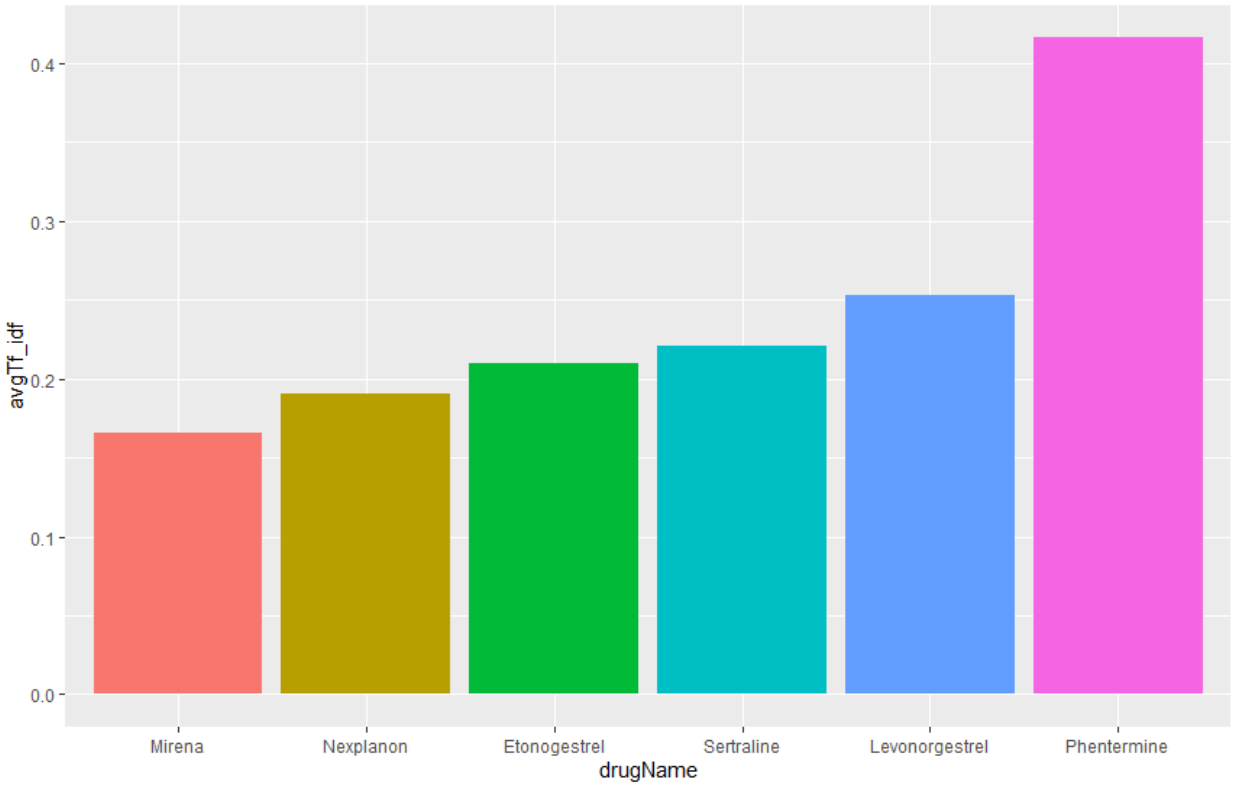
Wykres 14 Histogram współczynnika częstotliwości występowania słów(tf) w komentarzach dla 6 najczęściej komentowanych leków
Źródło: Opracowanie własne

Wykres 4 pokazuje, że dla każdego z 6 najbardziej ocenianych leków tj. Etonogestrel, Levonorgestrel, Nexplanon i Mirena (środki antykoncepcyjne), Phentermine (lek wspomagający odchudzanie) i Sertraline (środek antydepresyjny), zdecydowanie najwięcej jest słów mało powtarzających się. Oznacza to, że wśród komentarzy dla tego leku nie brakuje słów istotnych dla znaczenia całego zdania zgodnie z prawem Zipfa (Silge i Robinson, 2017).



Wykres 15: Histogram tf-idf termów dla 6 najczęściej ocenianych leków

Źródło: Opracowanie własne



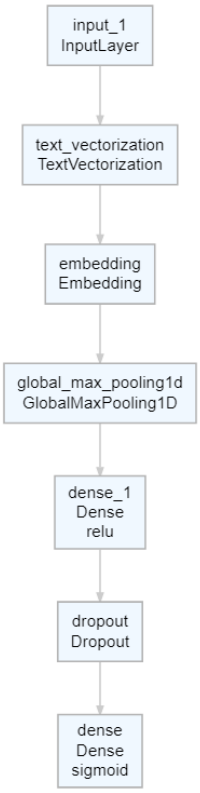
Wykres 16: Średni tf_idf dla 6 najczęściej ocenianych leków

Źródło: Opracowanie własne

Na histogramie tf-idf termów w komentarzach dla 6 najczęściej ocenianych leków dominują słowa o małej wadze (ponad 4000 termów przeciętnie dla zbadanego leku o tf-idf bliskim 0). To nie oznacza jednak, że zbiory tych komentarzy są nic nie znaczące. Na wykresie 5 widać, że dla każdego ze zbadanych leków jest grupa słów o współczynniku tf-df powyżej 1 czy nawet 2. Oznacza, że w zbiorze są słowa o dużej wadze, które pozwalają na zbadanie kontekstu wypowiedzi co jest kluczowe w możliwości stworzenia użytecznego modelu. Ma to odzwierciedlenie również w wykresie średnich współczynnika tf-idf (wykres 6), przy żadnym leku średnia tf_idf nie wynosi poniżej 0,1.

3.3 Wykorzystanie sieci neuronowej do oszacowania oceny leku na podstawie komentarzy przy użyciu biblioteki keras

Po dokonaniu analizy sentymentu, zbadaniu tf_idf można dojść do wniosku, że w zbiorze komentarzy znajdują się wyrazy o wysokiej wadze istotności(czyli takie o wysokim tf-idf) jak i bigramy (czyli n-gramy składające się z 2 słów) o dużej kontrybucji. Na tej podstawie można dojść do wniosku, że ten zbiór komentarzy nadaje się do skonstruowania modelu sieci neuronowej w celu zaklasyfikowania czy dany komentarz był opinią pozytywną bądź negatywną. Najpierw dokonano konwersji zmiennej oceny na zmienną binarną. Za ocenę negatywną przyjęto oceny od 0 do 6, zaś za pozytywne oceny od 7 do 10. W następnej kolejności zbudowano model sieci neuronowej. Jej architekturę przedstawiono na wykresie.

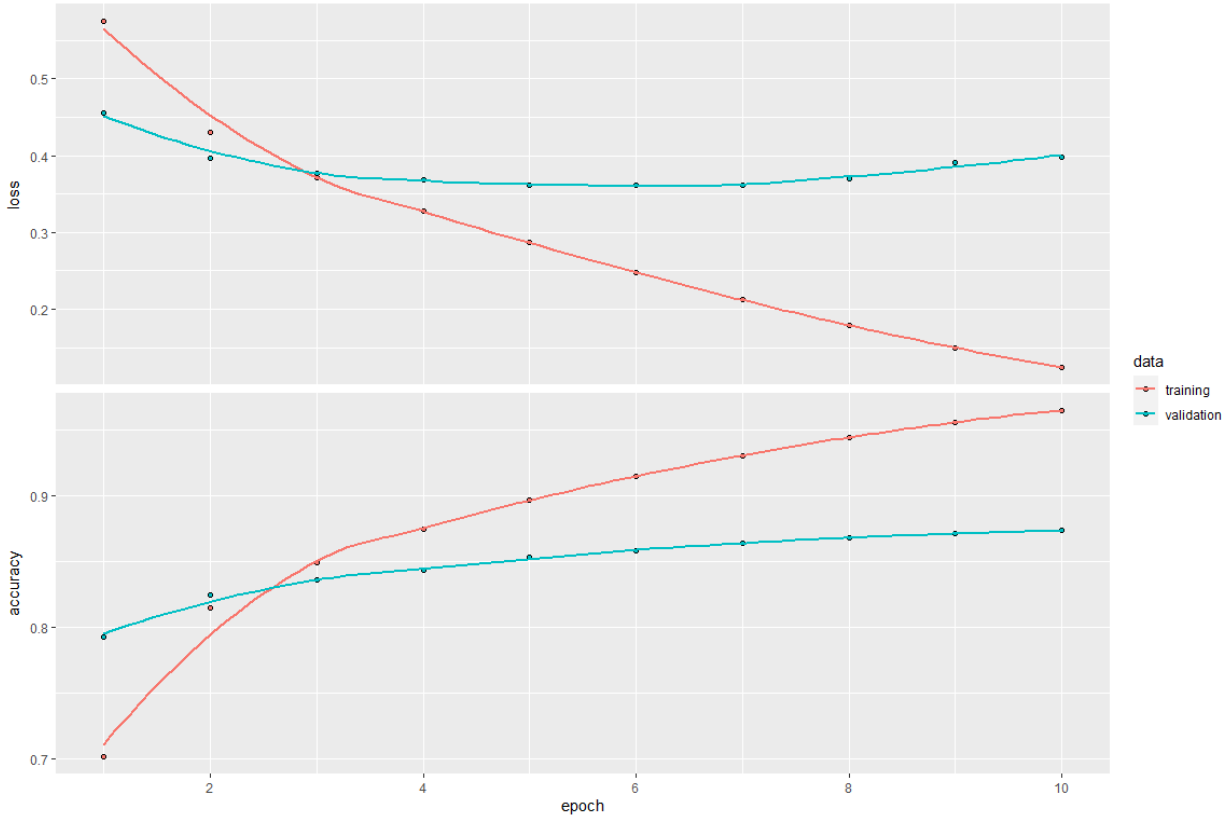


Wykres 17: Architektura modelu sieci neuronowej
Źródło: Opracowanie własne

W pierwszej kolejności stworzona jest warstwa wejściowa jednowymiarowa, następnie tworzona jest warstwa odpowiadająca za wektoryzację tekstu (Chollet, 2018). Tekst jest wówczas konwertowany na wektor bitów, gdzie jeżeli dane słowo wystąpi w danym tekście wejściowym, to wówczas wartość w indeksie tego słowa będzie wynosiła 1 i analogicznie 0, gdy tego słowa nie będzie. Znacznie ułatwia to wówczas przetwarzanie tekstów przez komputer. Drugą warstwą jest embedding, metoda powszechnie stosowana w NLP. Każde słowo jest konwertowane na wektor określonej długości. W kolejnej warstwa wbudowana po której jest warstwa jednowymiarowego max poolingu (pooling jest jednowymiarowy gdyż na wejściu znajdują się przekształcone w wektory słowa). Max pooling jest tutaj wskazany, gdyż chcemy się skupić na wyróżniających się wartościach w macierzach w przeciwieństwie do poolingu uśredniającego (Du i Shanker). W kolejnych fazach używana jest warstwa gęsta składająca się z 16 neuronów, w każdym z nich znajduje się funkcja aktywacji relu po której dokonywany jest dropout na poziomie 0.5, a na samym końcu znajduje się warstwa wyjściowa gęsta w której funkcją aktywacji jest funkcja sigmoidalną zwracająca prawdopodobieństwo, że komentarz jest pozytywny (Chollet, 2018).

Po zbudowaniu sieci poddano ją uczeniu na zbiorze treningowym. Dla znaczącego przyspieszenia procesu uczenia skorzystano z wsparcia GPU w bibliotece tensorflow oraz keras. Za ilość iteracji przejścia przez cały zbiór danych przyjęto 10, zaś za rozmiar batchowania 512. Wyniki uczenia w postaci wykresów skuteczności modelu oraz wartości funkcji straty na zbiorze

treningowym i walidacyjnym przedstawiona na wykresie 6.

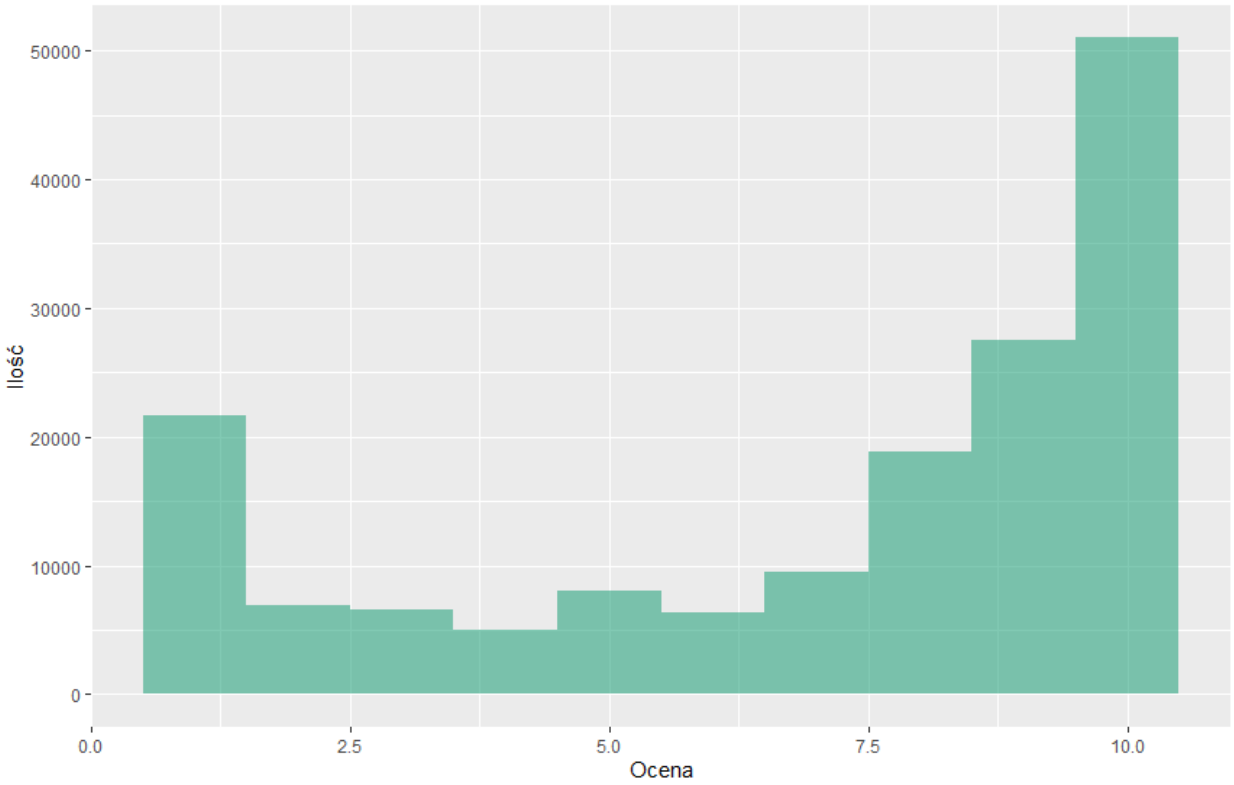


Wykres 18: Skuteczność i wartość funkcji straty modelu sieci neuronowej na zbiorze treningowym i walidacyjnym
Źródło: Opracowanie własne

Wartość funkcji straty na zbiorze walidacyjnym wyniosła 0,4 w 10 epoce, zaś dla zbioru treningowego wyniosła 0,05. Z kolei precyzja modelu dla zbioru walidacyjnego wyniosła ok. 0,85 a dla treningowego ponad 0,95.

3.4 Przedstawienie wyników analizy eksploracyjnej oraz estymacji wyników dokonanych przez model

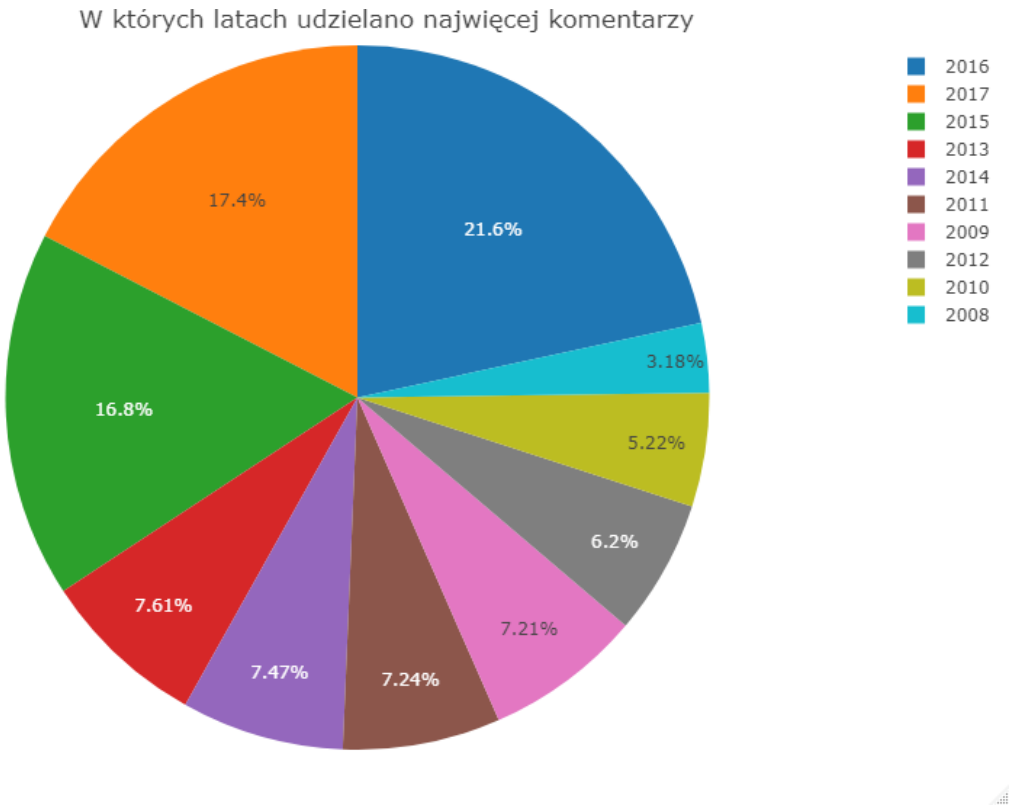
Do analizy eksploracyjnej użyto bibliotek graficznych takie jak: ggplot2 oraz plotly. Na początku zbadano histogram ocen leków. Na wykresie 5 przedstawiono jego wykres, który został stworzony przy pomocy biblioteki ggplot2 (Sievert, 2019) (Lander, 2017).



Wykres 19: Histogram ocen dla wszystkich leków

Źródło: Opracowanie własne

Według wykresu 7, najwięcej jest skrajnych ocen, czyli 1 oraz 9 i 10. Ta ostatnia ocena pojawia się najczęściej co oznacza, że większość opinii w zbiorze jest bardzo pozytywna i rekomendująca dany lek. W następnym kroku zbadano jak często oceniano leki w poszczególnych latach (2008-2017).



Wykres 20: Ilość komentarzy dla poszczególnych lat

Źródło: Opracowanie własne

Według wykresu 8 najwięcej komentarzy zebrano w roku 2016 – ponad 20 procent z całego zbioru. Oprócz tego dużo komentarzy napisano też w latach 2015 i 2017 (ponad 15%), w pozostałych latach wyniki już były poniżej 10%.

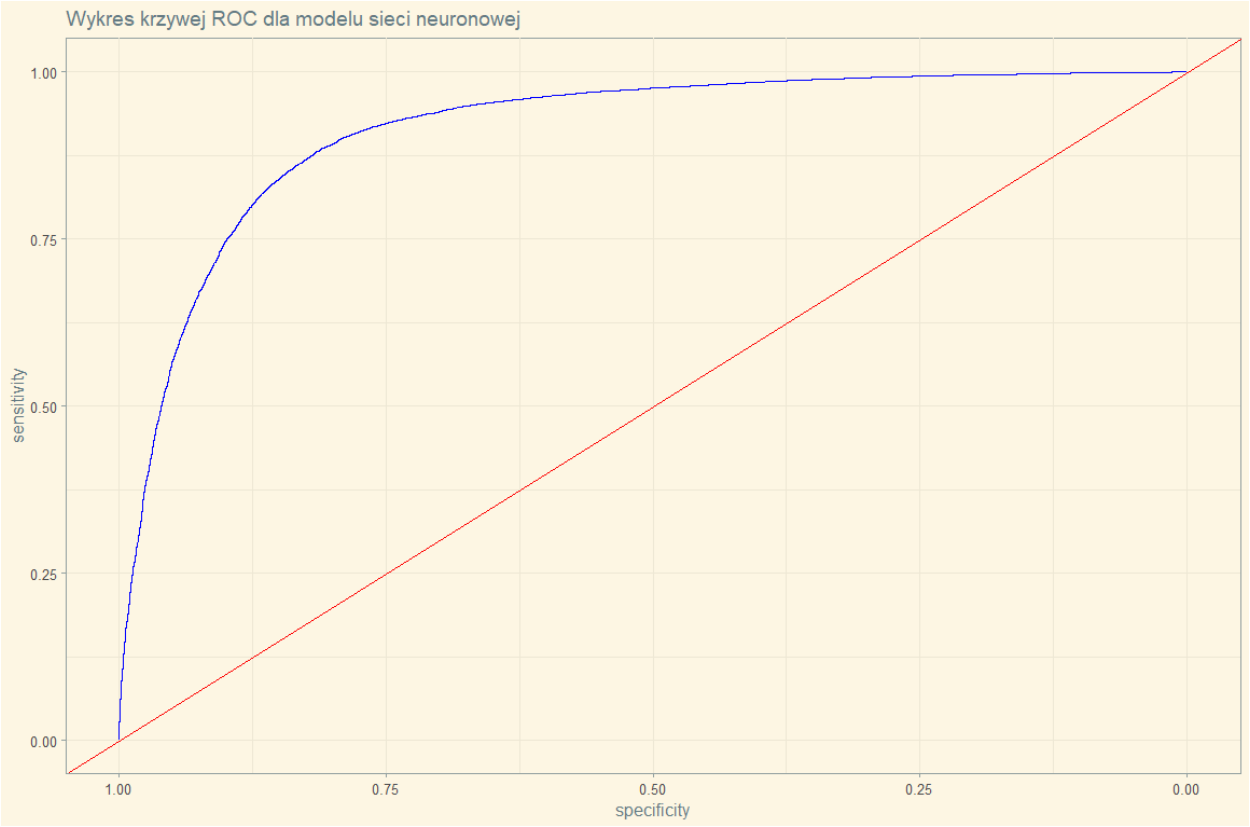
W kolejnym kroku zbudowano interaktywny dashboard za pomocą biblioteki shiny. Na wykresie 9 przedstawiono interaktywne wskaźniki KPI dla wybranej kategorii schorzeń czy dolegliwości. Ponadto w sekcji przedstawiono również wykres słupkowy dla najwyżej ocenianych leków w wybranej kategorii (Sievert, 2019).

Wykres 9 pokazuje, że leki na chorobę lokomocyjną były oceniane bardzo wysoko, średnia ocena wynosi aż ponad 8 przy ponad 200 komentarzach. Ponadto, komentarze te zebrały łącznie prawie 4000 polubień co wskazuje, że wiele osób poleca te środki. Do najwyżej ocenianych leków na tą dolegliwość należą Cyclizine, Marezine, Travel-Eze czy Dramamine: wszystkie z tych 4 leków

Wykres 21: Kluczowe informacje nt. komentarzy o lekach na chorobę lokomocyjną
Źródło: Opracowanie własne

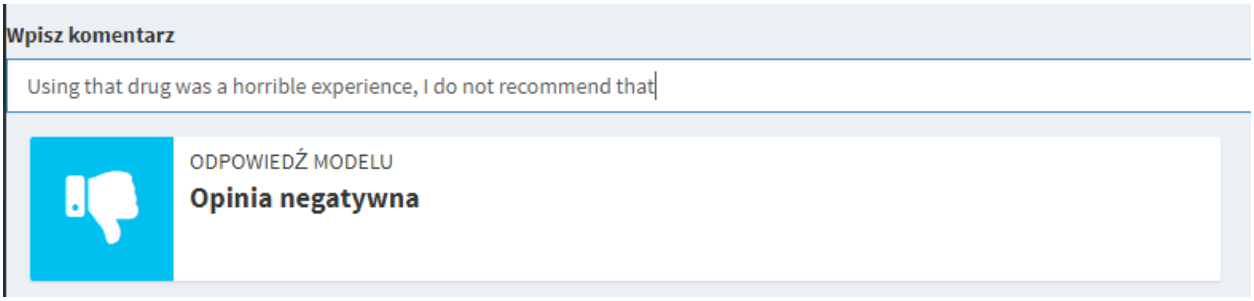
zbierały średnią ocenę ponad 9,5.

W następnej kolejności zaprezentowano wyniki klasyfikacji modelu, gdzie model sieci w zależności od podanego tekstu dokonywał weryfikacji czy opinia jest pozytywna bądź negatywna. Do zbadania dopasowania modelu do danych, wykorzystano bibliotekę pROC i caret.



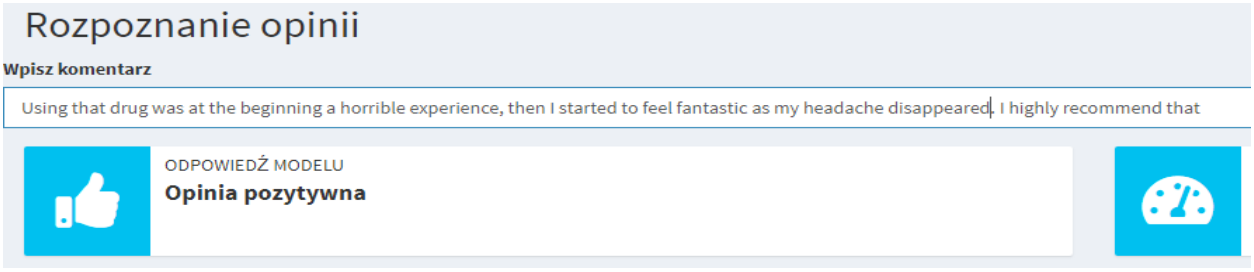
Wykres 22: Krzywa ROC modelu sieci neuronowej klasyfikującej komentarze
Źródło: Opracowanie własne

Pole pod krzywą ROC, która została pokazana na wykresie 10, wyniosło 0,91. Czułość wyniosła 0,88 a swoistość 0,81. Przełożyło się to na skuteczność modelu na poziomie 86% (Wickham, 2017). W ostatnim kroku spadano odpowiedź modelu na wprowadzony przez użytkownika tekst.



Wykres 23: Odpowiedź modelu na prosty komentarz

Źródło: Opracowanie własne



Wykres 24: Odpowiedź modelu na bardziej rozbudowany komentarz
Źródło: Opracowanie własne

Na początku sprawdzono odpowiedź modelu na prosty komentarz, który w dość jasny sposób wskazuje na negatywną opinię. Na wykresie 12 zaś ten komentarz rozbudowano, początek wskazuje wstępnie na negatywną opinię lecz dalsza część opinii pokazuje, że użytkownik był bardzo zadowolony z leku. Sieć udzieliła poprawnej odpowiedzi co wskazuje, że model ten poprawnie sobie radzi z rozbudowanymi zdaniami.

3.5 Użyte biblioteki oraz frameworki

Do stworzenia dashboardów, opracowania modelu, dokonania analizy eksploracyjnej oraz analizy NLP użyto następujących bibliotek:

- dplyr (biblioteka z pakietu tidyverse do pracy na ramkach danych)
- stringr (biblioteka z pakietu tidyverse do pracy na zmiennych tekstowych oraz korzystania z wyrażeń regularnych)
- ggplot2 (biblioteka z pakietu ggplot2 służąca do tworzenia wykresów)
- plotly (biblioteka służąca do tworzenia interaktywnych wykresów)
- tidytext (biblioteka do analizy NLP)
- wordcloud (biblioteka używana do obrazowania analizy NLP)
- keras (biblioteka do deep learningu będąca rozszerzeniem biblioteki tensorflow)
- caret (biblioteka do tworzenia i badania modeli uczenia maszynowego)

- pROC (biblioteka służąca do przeliczenia parametrów swoistości i czułości, pozwalająca na tworzenie krzywej ROC)
- shiny (framework służący do tworzenia interaktywnych i webowych dashboardów)
- shinydashboard (rozszerzenie frameworka shiny o dodatkowe komponenty i funkcjonalności)
- ggthemes (rozszerzenie biblioteki ggplot2 o dodatkowe style wykresów takie jak np. z tygodnika „The Economist” czy też z gazety „The Wall Street Journal”)
- DT (biblioteka do pracy z ramkami danych we frameworku shiny)

3.6 Kod źródłowy

Pełny kod źródłowy jest dostępny pod linkiem:
https://github.com/HomeSeeker88/MastersThesis
Cały projekt można sklonować przy pomocy programu git lub ściągnąć bezpośrednio z linku.

4. Podsumowanie wyników pracy

Na podstawie wyników dotyczących badań analizy sentymentu, tf-idf czy analizy eksploracyjnej zbudowano system rekomendacyjny oparty o model sieci neuronowej. W systemie tym można zbadać komentarze na temat każdego leku, który został oceniony w zbiorze, odnaleźć najlepiej oceniane leki na wybraną chorobę lub dolegliwość, czy też zobaczyć statystyki dopasowania modelu sieci neuronowej. Dzięki temu użytkownik może szybko znaleźć lek na dokuczające mu objawy oraz sprawdzić opinie innych na dany lek. W sekcji poświęconej modelowi można też zbadać jak poszczególne komentarze zostały rozpoznane przez model.

	uniqueID	review	rating	modelResponse	usefulCount	Correct
1	131683	"Sadly only lasted 5 days on Effexor XR. The side effects from the 75mg dose was unlike anything I have experienced before. Within 10 hours of the first dose I had severe anxiety - something I had never experienced before. Within hours of second dose the extreme nausea came on. By day 3 I was pratically bed-ridden in cold sweats and feeling completely "out of it" . I persevered hoping the effects would begin to subside but by the 5th day I had severe nausea, couldn't get out of bed from feeling so sick, horrible tinnitus (ringing in the ears) and felt spaced out to the point that I began to wonder if I would ever feel "normal" again. Advised by GP to cease immediately on hearing my side effects."	1	Opinia negatywna	18	true
2	131772	"Only the brand name version of Effexor works for me. Luckily my medical insurance agreed with my doctor and authorized me to continue getting the brand name version. I have found it to be very effective. An interesting "good side effect" is it helps with my serious face pain. A win win medicine."	10	Opinia pozytywna	28	true
3	132066	"Effexor XL has worked very well for me in combating my depression. I have been very diligent in taking the medicine properly and on schedule every day at near the same time. This past weekend due to an oops at the doctor's office a refill was not called in properly to the pharmacy before I went out of town for the weekend and I left town without any Effexor. Needless to say I experienced Effexor withdrawal symptoms. Heed my warning! It's not a good idea to miss a dose and you DEFINITELY DO NOT want to miss any doses. It was awful. Don't let the script run out."	9	Opinia pozytywna	66	true
4	131664	"Made depression worst ,was not effective even at 150 mg. I got an understanding of why people become alcoholics or drug abusers on Effexor/Venlafaxine. I would lay in bed all day waiting for life to pass by. I never had feelings of despair so severe. If I mised a dose or waited for an hour later to take Effexor I had an intense "Rush". I would pace around like on cocaine or speed. You cannot stop taking this medication. I am still trying to get off it! When I reduced my dose I had a drunk effect with nausea I just felt so awful. I consider this to be a drug that shoudnt be prescribed. Celexa and zoloft "robot drug"worked well on me for anxiety and depression but I do NOT recommend this medication. Going back to Elavil"	2	Opinia negatywna	25	true
5	131952	"After having a heart attack I became very depressed. I have been on Lexapro, Zoloft, Xanax, Ativan, Wellbutrin, and Buspar. None worked and I attempted suicide. I have Major depession and anxiety disorder, so take my comments in stride. The right drug for you is out there along with intensive therapy. I was switched to Effexor XR while in "the funny farm." After the typical waiting period for it to kick in, I can tell you I am a new man, looking at life like I used to. It has saved my life and my family. It hasn't done much for anxiety, so for the first few months after my severe "episode" I was on Seroquel. I have now been switched to clonazepam for that issue. Please consult a proper medical person."	10	Opinia pozytywna	39	true
6	131788	"I am a new person after taking this. I can live and not be in constant despair. It did take a while to work and together with praying the rosary, I have finally reached a state of contentment."	[illegible]	Opinia negatywna	45	false

Wykres 25: Tabela przedstawiające komentarze, ocenę oraz wynik modelu

W wykresie 15 na zielono zostały zaznaczone komentarze poprawnie ocenione przez model, błędnie ocenione opinie zostały zaś wyświetlone w kolorze czerwonym. Użytkownik ma też możliwość wpisać swój komentarz i sprawdzić odpowiedź sieci neuronowej. Model osiąga dokładność odpowiedzi na poziomie ok. 85% co można uznać za wynik satysfakcjonujący, model największe trudności ma z weryfikacją zdań w których występują zaprzeczenia. System można

byłoby rozszerzyć jeszcze o kolejne funkcjonalności:

- Rozszerzenie analizy komentarzy w innych językach europejskich takich jak niemiecki, polski czy francuski
- Oparcie systemu na oprogramowaniu służącym do przetwarzania bardzo dużej ilości danych jak np. Spark (biblioteki sparkr i sparklyr)

Taki bardziej rozbudowany system rekomendacji mógłby być wykorzystany przez np. sklepy internetowe, apteki czy też mogłyby być używane przez firmy farmaceutyczne w celu zbadania satysfakcji klientów.

5. Bibliografia

1. Chollet, F. (2018). *Deep Learing. Praca z językiem R i biblioteką Keras.*
2. Du, T. i Shanker, V. K. (brak daty). *Deep Learning for Natural Language Processing.*
3. Kallumadi, S. i Gräßer, F. (2018). *https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Drugs.com%29.*
4. Lander, J. (2017). *R dla każdego.*
5. Sievert, C. (2019). *Interactive Web-Based Data Visualization with R, plotly and shiny.*
6. Silge, J. i Robinson, D. (2017). *Text mining with R - A tidy approach.*
7. Technical University of Denmark. (2011). *http://www2.imm.dtu.dk/pubdb/pubs/6010-full.html.*
8. Wickham, H. (2017). *R for Data Science.*
9. Kibble, R. (2013). *Introduction to natural language processing*
10. Korbicz, J i Obuchowicz, A. i Uciński, D. (1994). *Sztuczne Sieci Neuronowe. Podstawy i Zastosowania*
11. Baheti, P. (2022). *12 types of Neural Network Activation functions. How to choose?*

OŚWIADCZENIE AUTORA PRACY DYPLOMOWEJ

1

LICENCJACKIEJ/MAGISTERSKIEJ

pod tytułem .....
.....
napisanej przez: .....**nr albumu** .....
pod kierunkiem .....

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca dyplomowa nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy dyplomowej jest identyczna z załączoną wersją elektroniczną.

Wyrażam zgodę na poddanie pracy dyplomowej kontroli, w tym za pomocą programu wychwytującego znamiona pracy niesamodzielnej, zwanego dalej programem, oraz na umieszczenie tekstu pracy dyplomowej w bazie porównawczej programu, w celu chronienia go przed nieuprawnionym wykorzystaniem, a także przekazanie pracy do Ogólnopolskiego Repozytorium Prac Dyplomowych.

Wyrażam także zgodę na przetwarzanie przez Szkołę Główną Handlową w Warszawie moich danych osobowych umieszczonych w pracy dyplomowej w zakresie niezbędnym do jej kontroli za pomocą programu oraz w zakresie niezbędnym do jej archiwizacji i nieodpłatnego udostępniania na zasadach określonych w zarządzeniu.

……. ……..
(data) (podpis autora)

1 Zastosować właściwe.