

Modelowanie głębokich przekonań na temat szczepień z użyciem sieci neuronowych

Autorzy:

Bogna Lew 184757,
Bartosz Strzelecki 184529,
Krzysztof Nazar 184698

Politechnika Gdańska

Wydział Elektroniki, Telekomunikacji i Informatyki

1. Wstęp

Polaryzacja społeczna to proces rozwarstwiania się społeczeństwa w taki sposób, że tworzą się dwie wyraźnie przeciwstawne bieguny społeczne.¹ Wzrost polaryzacji społecznej zwiększa się, co niesie za sobą pewne konsekwencje.² Z powodu pandemii ostatnim z zagadnień polaryzujących społeczeństwo jest chęć szczepień. Dzięki stworzeniu odpowiedniej bazy danych oraz sklasyfikowaniu jej zawartości możliwe jest wykorzystanie sztucznej inteligencji w celu analizy siatki pojęciowej danej grupy społecznej. Zebrane zostały teksty popierające oraz przeciwnie szczepieniom. Dzięki temu teksty zostały podzielone na dwie klasy reprezentujące przeciwstawne bieguny społeczne - autor tekstu wspiera lub jest przeciwny szczepieniom. Po zebraniu tekstów stworzono model przestrzeni wektorowej nad analizowanym zbiorem tekstów. Pozwoliło to na analizę siatki pojęciowej danej grupy. Wyniki zostały zwizualizowane za pomocą samoorganizującej sieci neuronowej - sieci Kohonena. Wykorzystano kilka sposobów analizy tekstów w celu wybrania najistotniejszych słów.

2. Opis algorytmów

2.1. Sztuczna sieć neuronowa

Sztuczna sieć neuronowa (ang. Artificial Neural Network) to system stworzony do przetwarzania informacji, którego struktura nawiązuje do budowy mózgu człowieka.³ Neurony tworzą kolejne warstwy sieci, połączone są synapsami. W celu zrozumienia mechanizmu działania sieci neuronowej, poniżej wyjaśnione zostały najważniejsze pojęcia.

2.2. Sztuczny neuron

Sztuczny neuron, czyli perceptron, to atomowy element sieci neuronowej. Każdy perceptron posiada wejścia przyjmujące wartości rzeczywiste lub binarne, będące odpowiednikami dendrytów.⁴ Sztuczny neuron to jednostka obliczeniowa, która na podstawie określonej funkcji aktywacji oblicza i podaje jako wynik pewną wartość na podstawie sumy ważonych danych wejściowych. Parametry wewnętrzne

¹ <https://encyklopedia.pwn.pl/haslo/polaryzacja-spoeczna;3959320>

² <https://ptsp.pl/rosnie-polaryzacja-spoeczna-megatrendy-2050/>

³ <https://www.ibm.com/pl-pl/cloud/learn/neural-networks>

⁴ <https://edu.pjwstk.edu.pl/wyklady/nai/scb/wyklad2/w2.htm>

perceptronu to n wag połączeń, odpowiadających wagom synapsów łączących naturalne neurony oraz wartość odchylenia b (ang. bias) pozwalająca uzyskać nieliniowe przekształcenie danych wejściowych na wyjście.⁵

2.3. Sposób działania sztucznego neuronu

Założmy, że i -ty neuron ma przypisaną wagę w_i . Dla danych wejściowych x_1, \dots, x_n należy obliczyć sumę ważoną s , biorąc pod uwagę wartość odchylenia b .

$$s = \sum_{i=1}^n x_i w_i + b$$

Należy przyjąć wartość progową p , czyli parametr perceptronu wyznaczający stałą granicę, powyżej której perceptron jest wzbudzony (tzn. na wyjściu ma wartość 1). Wyjście y będzie równe 1, gdy wartość sumy ważonej s będzie większa lub równa p . W przeciwnym przypadku wartość wyjścia y równa się 0.

2.4. Sposób działania perceptronu

Algorytm uczenia perceptronu polega na automatycznym doborze wag na podstawie podanych przykładów wejściowych. Algorytm wygląda następująco:

1. Wagi inicjowane są losowo
2. Dla każdego z podanych przykładów obliczana jest odpowiedź perceptronu
3. Jeśli perceptron podał nieprawidłową odpowiedź, jego wagi są modyfikowane na podstawie współczynnika uczenia n , oczekiwanej odpowiedzi d oraz odpowiedzi neuronu y , na przykład w ten sposób:

$$w_1 = w_1 + n \cdot (d - y) \cdot x_1$$

$$w_2 = w_2 + n \cdot (d - y) \cdot x_2$$

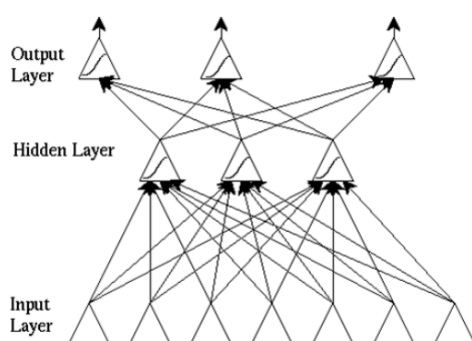
$$b = b + n \cdot (d - y)$$

Proces uczenia na podanych przykładach przebiega dopóki następują jakiegokolwiek zmiany wag połączeń.

⁵ <https://home.agh.edu.pl/~horzyk/lectures/biocyb/BIOCYB-SieciNeuronowe.pdf>

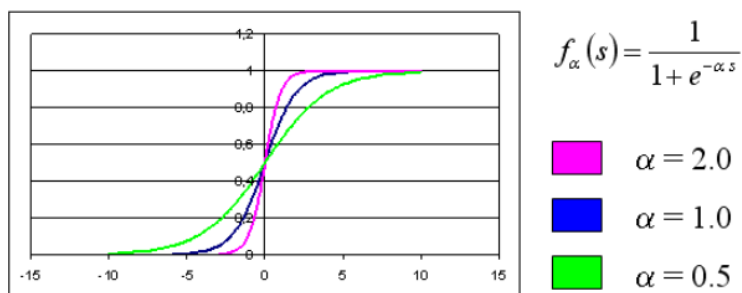
2.5. Warstwy sieci neuronowej

Sieć neuronowa składa się z wielu warstw, które można podzielić na trzy rodzaje: wejściowa (ang. Input Layer), niewidoczna (ang. Hidden Layer), wyjściowa (ang. Output Layer). Wyjścia neuronów należących do warstwy niższej połączone są z wejściami neuronów należących do warstwy wyższej (każdy z każdym). Działanie sieci neuronowej polega na liczeniu odpowiedzi neuronów w kolejnych warstwach. Rozpoczyna się od podania do pierwszej warstwy sygnałów wejścia, a następnie na ich podstawie wyznaczane są odpowiedzi drugiej warstwy neuronów, itd. Jako wyjście traktowane są odpowiedzi neuronów z ostatniej warstwy.



Metoda projektowania architektury sieci neuronowej⁶

W celu optymalnego wykorzystania struktury sieci neuronowej funkcja aktywacji neuronów nie powinna być skokowa. Można wykorzystać wszelkie funkcje pamiętając, żeby były monotoniczne oraz różniczkowalne. Często stosowaną funkcją aktywacji jest funkcja sigmoidalna.



$$\lim_{\alpha \rightarrow 0} f_{\alpha}(s) = 0.5 \quad \lim_{\alpha \rightarrow +\infty} f_{\alpha}(s) = \begin{cases} 1 & \Leftrightarrow s > 0 \\ 0.5 & \Leftrightarrow s = 0 \\ 0 & \Leftrightarrow s < 0 \end{cases}$$

⁶ <https://edu.pjwstk.edu.pl/wyklady/nai/scb/wyklad3/w3.htm>

2.6. Algorytm propagacji wstecznej

Ogromną zaletą sieci neuronowych jest to, że wag nie trzeba ustawiać ręcznie, ale można je wytrenować. Jedną z metod pozwalających na znajdowanie w przybliżeniu optymalnych wag jest metoda wstecznej propagacji błędu. Wynik obliczeń dla neuronów w wyższych warstwach bazowany jest na wynikach z poprzednich. Więc, w algorytmie uczenia na podstawie otrzymanego wyniku w warstwie wyjściowej należy odpowiednio zmodyfikować rekurencyjnie wszystkie poprzednie warstwy poruszając się od warstwy wyjściowej do wejściowej. Skala modyfikacji neuronu będzie proporcjonalna do rozmiaru błędu obliczeń dla tego neuronu.

Schemat trenowania sieci wygląda następująco:

1. Ustalana jest topologia sieci - liczba warstw, liczba neuronów w każdej warstwie.
2. Wagi inicjowane są losowo.
3. Dla danych wejściowych obliczane są odpowiedzi sieci (warstwa po warstwie).
4. Różnica pomiędzy wynikiem y od danego neuronu oraz poprawną odpowiedzią t interpretowana jest jako błąd dla danego neuronu.
5. Błędy propagowane są do wcześniejszych warstw.
6. W każdym neuronie w sieci modyfikowane są wagi na podstawie wartości błędu oraz wielkości przetwarzanych w tym kroku sygnałów.
7. Dla kolejnych wektorów uczących schemat powtarzany jest od punktu 3.
8. Trenowanie kończone jest gdy średni błąd na danych treningowych przestanie maleć.

Zwykle błąd d (różnica pomiędzy wynikami obliczeń y oraz poprawną odpowiedzią t) liczony jest za pomocą poniższego wzoru:

$$d(w_1, \dots, w_k) = \frac{1}{2} (f(w_1 z_1 + \dots + w_k z_k) - t)^2$$

gdzie:

⁷ <https://edu.pjwstk.edu.pl/wyklady/nai/scb/wyklad3/w3.htm>

w_1, \dots, w_k - wagi połączeń wchodzących

z_1, \dots, z_k - sygnały z poprzedniej warstwy wchodzącego do neuronu

t - poprawna odpowiedź obliczeń neuronu

Aby ustalić o jaką wartość powinna zmienić się waga neuronu, błąd całkowity powinien być rozłożony na połączenia wprowadzającego sygnały do danego neuronu. Składowa błędu dla każdego i -tego połączenia określenia jest jako pochodna cząstkowa błędu względem i -tej wagi.⁸

2.7. Sieć Kohonena

Samoorganizujące się mapy (Self Organizing Maps, SOM), czyli sieci Kohonena, to sieci, których zadaniem jest organizacja wielowymiarowej informacji w taki sposób, aby można było ją prezentować i analizować w przestrzeni o znacznie mniejszej liczbie wymiarów, np. mapie na dwuwymiarowym ekranie. Podobne dane wejściowe powinny być blisko położone względem siebie na mapie.⁹ Uczenie SOM polega na zmianach współrzędnych neuronów, tak, aby jak najlepiej odzwierciedlały wzorzec zgodny ze strukturą analizowanych danych. Sieci Kohonena dopasowują swoją strukturę do analizowanego zbioru danych.¹⁰

2.8. Dane wejściowe

Zebrane zostały teksty napisane w języku angielskim popierające oraz będące przeciwne szczepieniom. Każdemu z tekstów został przypisany odpowiedni znacznik w zależności od tego czy autor tekstu wspiera lub jest przeciwny szczepieniom. Dzięki temu teksty zostały podzielone na dwie klasy reprezentujące przeciwstawne bieguny społeczne.

⁸ <https://edu.pjwstk.edu.pl/wyklady/nai/scb/wyklad3/w3.htm>

⁹ <https://edu.pjwstk.edu.pl/wyklady/nai/scb/wyklad4/w4.htm>

¹⁰ https://en.wikipedia.org/wiki/Self-organizing_map

Po zebraniu tekstów stworzono model przestrzeni wektorowej nad analizowanym zbiorem tekstów. Pozwoliło to na analizę siatki pojęciowej danej grupy. Sprawdzono jak słowa w obrębie tych dwóch klasach tekstów wiążą się ze sobą. Każdy tekst opisany jest zbiorem słów które występują w tym tekście.

Następnie zwizualizowna wyniki za pomocą samoorganizującej sieci neuronowej - sieci Kohonena.

Następnie zobaczymy jak bardzo oraz w jaki sposób możemy modyfikować tą siatkę w zależności od dodawania nowych danych. Można zmienić jej algorytm uczący i sprawdzimy jaki wpływ będzie to miało na organizację danych.

Ważenie binarne - wektor o długości liczbie unikalnych słów które występują we wszystkich tekstach(w wierszu) w kolumnach są zadane teksty. na przecięciu będzie wartość 0 lub 1 w zależności czy dany tekst zawiera dane słowo. Różne rodzaje ważenia: [TF IDF](#). Do stworzenia modelu warto wykorzystać zaimplementowaną bibliotekę.

Model podajemy na wejście do sieci samoorganizującego - do kohonena . patrzemy jak te teksty się rozdzielają - czy potrafimy rozdzielić teksty. Jeżeli nie możemy manipulować parametrami w celu uzyskania prawidłowych

2.9. Algorytm ważenia - TF-IDF

TF-IDF to skrót od Term Frequency — Inverse Document Frequency. Jest to technika analizy liczby wystąpień danych wyrazów w tekście. Każdemu z wyrazów przypisany jest wynik informujący o istotności tego słowa w analizowanych danych wejściowych.

3. Analiza danych

W badaniu wykorzystano zaimplementowaną bibliotekę Minisom.¹¹ Biblioteka posiada zaimplementowane sieci samoorganizujące oparte na bibliotece Numpy.¹² W celu zrozumienia wyniku otrzymanego dzięki wykorzystaniu biblioteki Minisom warto przybliżyć kluczowe pojęcia związane ze sposobem analizy danych.

3.1. Macierz błędu

Podstawowym narzędziem wykorzystywanym do oceny jakości klasyfikacji jest macierz błędu (ang. Confusion matrix). W przypadku klasyfikacji binarnej w tym badaniu występuje podział na dwie klasy:

- Positive - tekst nie popiera szczepień
- Negative - tekst popiera szczepienia

		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	TP True Positive	FP False Positive
	negatives	FN False Negative	TN True Negative

Macierz błędu.¹³

Na powyższej grafice można wyróżnić 4 ćwiartki:

- True Positives (TP) - wszystkie elementy klasy True poprawnie sklasyfikowane przez sieć jako elementy True
- False Positives (FP) - wszystkie elementy klasy False niepoprawnie sklasyfikowane przez sieć jako elementy True

¹¹ <https://github.com/JustGlowing/minisom>

¹² <https://numpy.org/>

¹³ <https://mathspace.pl/matematyka/confusion-matrix-macierz-bledu-tablica-pomylek-czyli-ocena-jakosci-klasyfikacji-czesc-1/>

- False Negatives (FN) - wszystkie elementy klasy True niepoprawnie sklasyfikowane przez sieć jako elementy False
- True Negatives (TN) - wszystkie elementy klasy False poprawnie sklasyfikowane przez sieć jako elementy False

Korzystając z biblioteki scikit learn w Pythonie, użytkownik może wydrukować raport klasyfikacji (ang. Classification report), którego zawartość oparta jest na macierzy błędów.

	precision	recall	f1-score	support
False	0.60	0.50	0.55	6
True	0.73	0.80	0.76	10
accuracy			0.69	16
macro avg	0.66	0.65	0.65	16
weighted avg	0.68	0.69	0.68	16

Przykład Classification report.

Każda z kolumn pozwala ocenić jakość klasyfikacji.

3.2. Wsparcie (ang. Support)

Wartość Support informuje o tym, ile losowo wybranych przykładów z danej kategorii zostało wykorzystane do grupy testowej. Powyższy przykład pokazuje, że algorytm był testowany na 16 tekstach: 6 popierające szczepionki oraz 10 nie popierających szczepionki.

3.3. Dokładność (ang. Accuracy)

Dokładność to metryka informująca o stosunku poprawnie dokonanych klasyfikacji do ilości wszystkich dokonanych klasyfikacji.¹⁴ Dokładność warto brać pod uwagę, gdy w analizowanych danych ilość elementów należących do innych klas jest do siebie jak najbardziej zbliżona. W przeciwnym wypadku wartość dokładności może być mało miarodajna. Oczekuje się jak najwyższej dokładności. Jeśli dokładność

¹⁴ <https://www.numpyninja.com/post/recall-specificity-precision-f1-scores-and-accuracy>

wynosi 1.00 algorytm nie pomylił się ani razu w dopasowywaniu danych. Dokładność D można obliczyć za pomocą wzoru:

$$D = \frac{TP + TN}{TP + TN + FP + FN}$$

3.4. Precyzja (ang. Precision)

Precyzja to metryka informująca o tym, ile wśród przykładów zaprognozowanych pozytywnie jest rzeczywiście pozytywnych.¹⁵ Precyzję wyznacza się jako liczbę prawidłowo zaprognozowanych pozytywnych wartości (TP) podzielonej przez sumę wszystkich zaprognozowanych pozytywnie. Wartość precyzji powinna przyjmować wartość jak najbliższą 1.00. Precyzję P można obliczyć za pomocą wzoru:

$$P = \frac{TP}{TP + FP}$$

3.5. Czułość (ang. Recall)

Ta metryka informuje o tym jaki jest udział prawidłowo przewidzianych przypadków pozytywnych (TP) wśród wszystkich przypadków pozytywnych (bierze się również pod uwagę przypadki które błędnie zostały zaklasyfikowane do negatywnych). Warto zauważyć, że jeśli algorytm nie rozpozna żadnego pozytywnego przypadku błędnie (FN równe 0), to wartość czułości będzie wynosić 1. Czułość jest parametrem, który powinien dążyć do jedynki. Wartość Czułości C można obliczyć za pomocą wzoru:

$$C = \frac{TP}{TP + FN}$$

3.6. F1-Score

Ta metryka to średnia harmoniczna pomiędzy czułością (ang. recall) oraz precyzją (ang. precision). Im bliższa wartość tego parametru jest bliższa jedynki, tym lepszy algorytm klasyfikujący.¹⁶ W przypadku idealnej czułości i precyzji wartość tego parametru wynosi 1. F1-Score używa się gdy precyzja i czułość są interpretowane jako równie istotne. Wartość F1-Score F można obliczyć za pomocą wzoru:

¹⁵ <https://ksopyla.com/data-science/precision-recall-f1-miary-oceny-klasyfikatora/>

¹⁶ <https://www.statystyczny.pl/macierz-bledow-raport-dokladnosc-czulosc-precyzja/>

$$F = \frac{2TP}{2TP + FP + FN}$$

3.7. Macro average oraz weighted average

Macro avg to wartość średniej arytmetycznej z danego parametru dla każdej kategorii. Weighted avg to wartość średniej ważonej biorąc pod uwagę liczbę przypadków z każdej kategorii.

Gdy pomiędzy wartościami tymi dwóch średnich występuje znacząca różnica, wtedy do zbory danych testowych wylosowane zostały nieproporcjonalne ilości elementów z danych kategorii.

4. Wynik eksperymentu

4.1. Dane wejściowe

Dane wejściowe to 146 tekstów w języku angielskim: 72 popierające szczepienia oraz 74 przeciwnych szczepieniom. Teksty oraz ich źródła i przynależność do klasy zostały opisane w pliku w formacie json.¹⁷ Przy tworzeniu zmiennej zawierającej teksty oraz ich klasę, usuwane są niektóre słowa. Zostało to wykonane za pomocą metod zaimplementowanych w bibliotece NLTK (Ang. Natural Language Tagging Toolkit).¹⁸ Natural Language Processing pozwala na określenie do jakiej części mowy należy dane słowo (Ang. Part of Speech Tagging). Z danych wejściowych usunięto cyfry oraz znaki specjalne, ponieważ założono, że nie mają one istotnego wpływu na to, czy dany tekst popiera lub nie popiera szczepień. Uzyskane dane zostały podzielone funkcją z biblioteki Skleran w następujący sposób: 75% - dane treningowe, 25% dane testowe.¹⁹

Za pomocą funkcji zaimplementowanych w bibliotece NLTK, w zbiorze analizowanych słów zostały tylko rzeczowniki, przymiotniki i przysłówki.

¹⁷ <https://docs.fileformat.com/web/json/>

¹⁸ <https://www.nltk.org/>

¹⁹ https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

4.2. Permutation feature importance

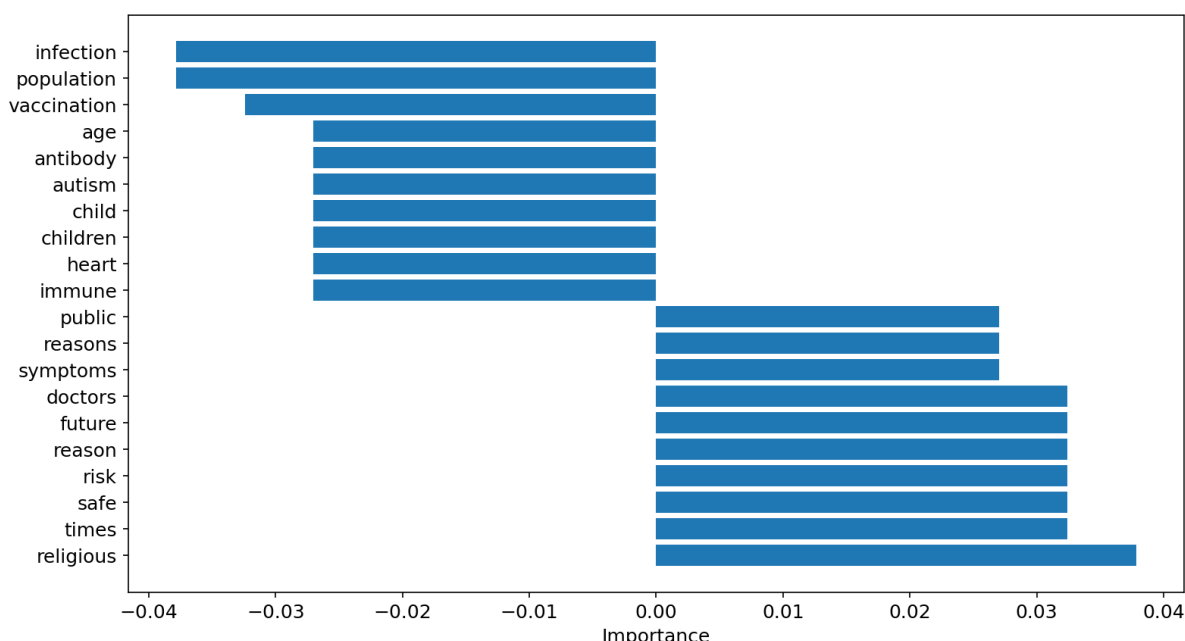
Analiza tekstów rozpoczęła się od zaimplementowanie metody poszukiwania najbardziej istotnych słów w tekstach. W tym celu została wykorzystana metoda sprawdzająca znaczenie permutacji danego słowa, czyli permutation importance (Ang. Permutation feature importance, PFI).²⁰ Znaczenie permutacji jest techniką szczególnie przydatną w przypadku nieliniowych estymatorów. Wartość znaczenia permutacji obliczona jest po wytrenowaniu danego modelu. Metoda ta polega na losowym wymieszaniu kolumny danych. Następnie badany jest spadek dokładności klasyfikatora. Im większy spadek dokładności, tym ważniejsza jest konkretna dana w modelu. Taka procedura pozwala określić jak bardzo model danych zależy od danego słowa.

4.3. Klasyfikator Keras

Na początku został zaimplementowany klasyfikator Keras (Ang. Keras Classifier), który przetrenowano na danych treningowych. Klasyfikator Keras to sieć neuronowa z odpowiednią ilością warstw ukrytych. Wykorzystana została gęsta sieć neuronowa typu feed-forward (FF-NN). Sieć zbudowana jest z 1 warstwy wejściowej, 3 warstw ukrytych oraz 1 warstwy wyjściowej. Funkcją aktywacji w neuronach jest sigmoid. Do neuronów wprowadzono tzw. wektor bias, który powoduje wzrost ich możliwości uczenia się.²¹ Dzięki temu, możliwe jest przesunięcie progu aktywacji w zależności od wagi biasu. Należy pamiętać, że zwiększa to ilość dodatkowych obliczeń. Wytrenowany model klasyfikatora został podany na wejście funkcji wyznaczającej Permutation Importance. Wykres przedstawiający 20 najważniejszych słów został przedstawiony poniżej.

²⁰ https://scikit-learn.org/stable/modules/permutation_importance.html

²¹ <http://galaxy.agh.edu.pl/~vlsi/Al/bias/bias.html>



Permutation importance dla danych otrzymanych na podstawie wytrenowanego modelu klasyfikatora Keras.

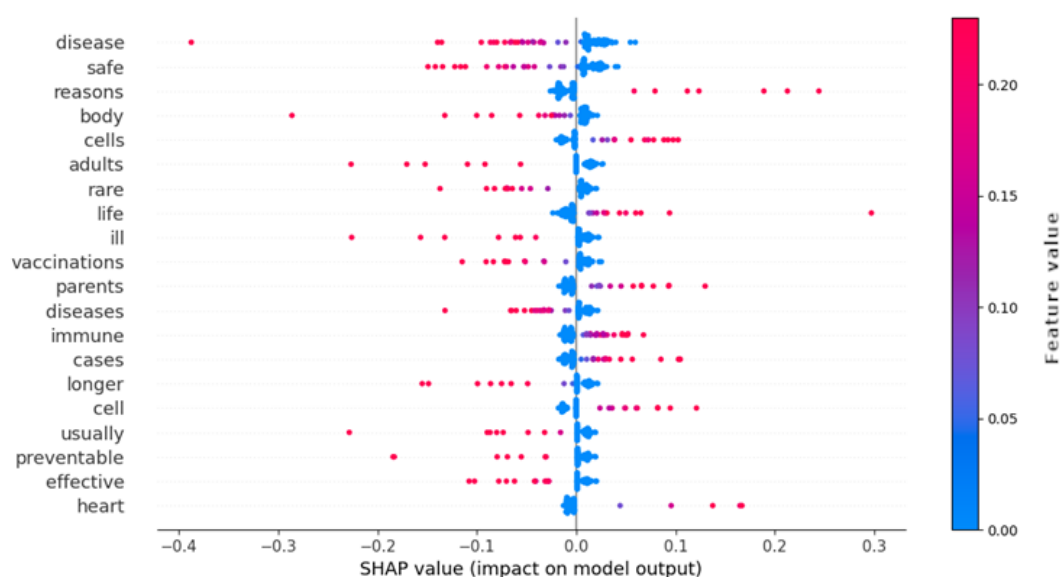
Wykres dla 20 najważniejszych słów wygląda bardzo symetrycznie. W obydwu grupach tekstów znajdują się słowa o równej wartości Permutation Importance. Słowa, które są najważniejsze dla tekstów popierających szczepienia to: infection, population oraz vaccination. Natomiast dla przeciwnej klasy tekstów są to: religious, times oraz safe.

Ten sam model klasyfikatora posłużył do implementacji kolejnej metody szukania kluczowych słów. Do jej implementacji została wykorzystana biblioteka SHAP (Ang. SHapely Additive exPlanations).²² Wartości SHAP można użyć do identyfikacji i wizualizacji ważnych relacji w modelu danych. W tym przypadku wartość SHAP pozwala określić, w jakim stopniu każdy czynnik w analizowanym modelu przyczynił się w procesie uczenia.

Każde wystąpienie danego słowa reprezentowane jest przez pojedynczą kreskę w danym wierszu. Położenie kropki względem poziomej osi jest określone przez wartość SHAP dla danego słowa. Im większa odległość od linii pionowej o wartości 0.0, tym większy wpływ tego słowa na klasyfikację pojedynczego artykułu. Gdy słowo pojawia się w wielu artykułach i jest w nich tak samo istotne, kropki gromadzą

²² <https://shap.readthedocs.io/en/latest/index.html>

się w danym wierszu, tworząc gęste, pionowe kreski. Kolor kropek służy do reprezentowania tzw. wartości cechy (Ang. Feature value).²³ Jest to informacja jak bardzo znaczące było wystąpienie danego słowa w danym artykule względem wszystkich obserwacji dla danego słowa. Słowa, których wartości SHAP mają wartość ujemną, pochodzą od tekstów, które nie są antyszczepionkowe. Słowa są posortowane malejąco względem bezwzględnej wartości całkowitej SHAP - im wyższa pozycja na wykresie, tym ważniejsze jest to słowo w procesie nauki sieci. Na poniższym wykresie zostało przedstawione 20 najbardziej istotnych słów.



Wykres przedstawiający wartość SHAP 20 najbardziej istotnych słów.

W tym przypadku słowa disease, safe, reasons to trzy słowa o największym wpływie na proces nauki. Słowa, które były najbardziej kluczowe w klasyfikacji tekstów popierających szczepienia to: disease, save oraz body. Natomiast dla tekstów antyszczepionkowych trzy najważniejsze słowa to: reasons, cells, life. Wartości SHAP uzyskane dla poszczególnych słów mają podobną charakterystykę. Widoczna jest duża liczba czerwonych kropek z jednej strony pionowej linii z wartością 0.0, a po jej przeciwnej stronie znajdują się gęste niebieskie kropki. Na wykresie jest pokazana tylko część najbardziej znaczących słów i taka charakterystyka pokazuje, że te słowa rzeczywiście przyczyniły się do klasyfikacji danego tekstu do odpowiedniej klasy.

²³ <https://www.data-cowboys.com/blog/shap-values-and-feature-variance>

Należy pamiętać, że wartość SHAP pozwala ocenić istotność danego słowa podczas procesu nauki sieci, ale nie ocenia jakości końcowego efektu nauki. Z tego powodu, kolejnym etapem analizy było sprawdzenie jak dobrze sieć nauczyła się klasyfikować teksty na podstawie dostarczonych danych.

Do określenia jak dobrze klasyfikator nauczył się rozpoznawać teksty, wykorzystano wbudowane funkcje z biblioteki Keras.²⁴ Zostały wykonane predykcje na podstawie danych treningowych oraz testowych modelu. Prawdziwą informację o tym jak dobrze nauczyła się sieć niesie za sobą predykcja dla danych testowych. Poniżej przedstawiony został raport klasyfikacji (Ang. classification report) dla uzyskanego modelu klasyfikatora.

----- Evaluation on Training Data -----					
	precision	recall	f1-score	support	
False	0.98	0.94	0.96	53	
True	0.95	0.98	0.96	56	
accuracy			0.96	109	
macro avg	0.96	0.96	0.96	109	
weighted avg	0.96	0.96	0.96	109	
----- Evaluation on Test Data -----					
	precision	recall	f1-score	support	
False	1.00	0.89	0.94	19	
True	0.90	1.00	0.95	18	
accuracy			0.95	37	
macro avg	0.95	0.95	0.95	37	
weighted avg	0.95	0.95	0.95	37	

Raport klasyfikacji (Ang. Classification report)

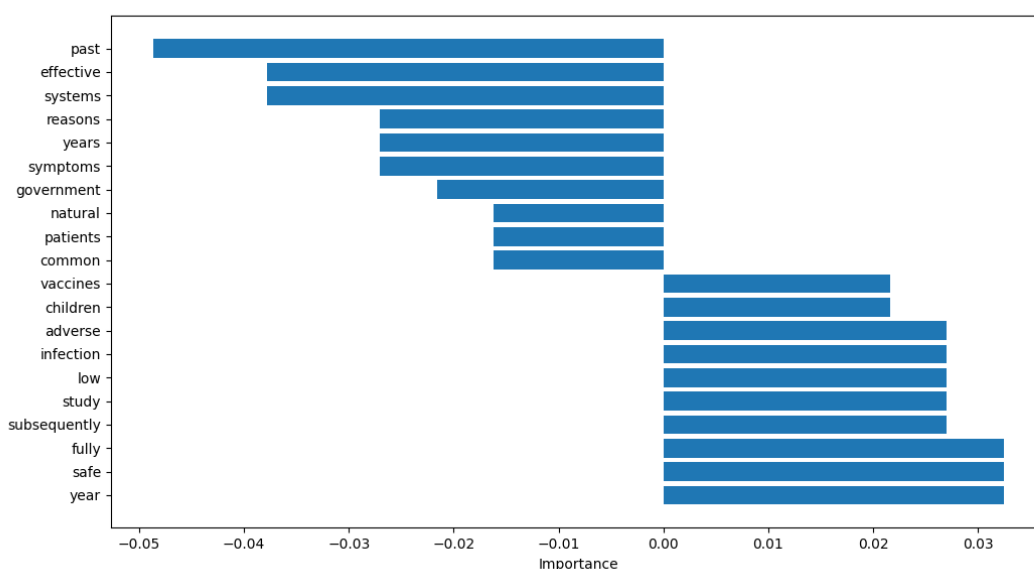
Wyniki predykcji są bardzo satysfakcjonujące. Wartości wyników dla obydwu typów danych są do siebie bardzo zbliżone. Jedną z najważniejszych wartości jest accuracy określająca dokładność predykcji. Danych treningowych wartość accuracy wynosi 96%, a dla danych testowych 95%. Wyniki macierzy błędów sugerują, że sieć nauczyła się znakomicie rozpoznawać i klasyfikować teksty.

²⁴ <https://www.educba.com/keras-model-predict/>

4.4. Klasyfikator SVM

Aby wykonać porównanie jakości działania klasyfikatorów, został zaimplementowany klasyfikator SVM (Ang. support-vector machine). Support Vector Machine to nadzorowany algorytm klasyfikacji, którego zadaniem jest jak najlepsze oddzielenie elementów należących do dwóch grup. W problemach klasyfikacji tekstów najczęściej wykorzystuje się SVM z jądrem liniowym. Oznacza to, że dane należące do oddzielnych klas można oddzielić pojedynczą linią.

Podobnie jak w przypadku klasyfikatora Kerasa, ten klasyfikator także został wytrenowany na danych treningowych, a następnie podany na wejście funkcji określającej Permutation Importance. Poniżej zamieszczony został wykres zawierający 20 najbardziej istotnych słów.



20 najbardziej istotnych słów wybranych na podstawie Permutation Importance.

Powyższy wykres jest zbliżony do poprzedniego. W tym wypadku widoczna jest mniejsza symetria pomiędzy poszczególnymi stronami wykresu. Biblioteka w której został zaimplementowany SVM pozwala na ocenę jakości wytrenowanego modelu. W tym celu tworzy się raport klasyfikacji przedstawiający między innymi dokładność klasyfikatora. Raport klasyfikacji dla tego klasyfikatora został umieszczony poniżej.

	precision	recall	f1-score	support
False	0.79	0.71	0.75	21
True	0.67	0.75	0.71	16
accuracy			0.73	37
macro avg	0.73	0.73	0.73	37
weighted avg	0.74	0.73	0.73	37

Raport klasyfikacji (Ang. classification report) na podstawie SVM.

Dokładność równa 75% sugeruje, że klasyfikator nauczył się poprawnie rozpoznawać teksty. Precyzja rozpoznawania jest nieznacznie wyższa dla tekstów popierających szczepienia.

Na podstawie powyższej analizy, stwierdzono, że w dalszej części rozważań wykorzystywany będzie klasyfikator SVM. Według raportów klasyfikacji osiąga on niższą dokładność od klasyfikatora Keras, jednak pozwala w dokładniejszy sposób zobaczyć w jaki sposób dokonywana jest klasyfikacja.

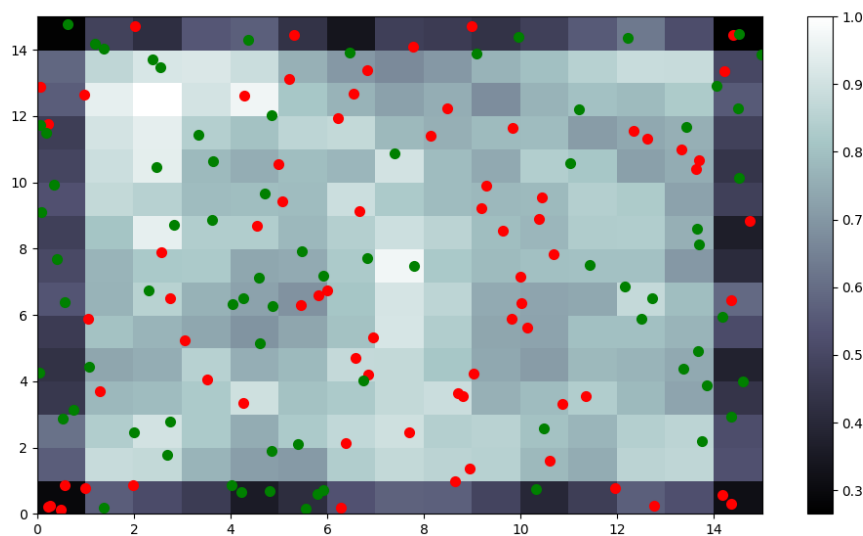
4.5. Samoorganizująca się mapa Kohonena

Następnie w celu wizualizacji klasyfikacji zaimplementowana została samoorganizująca się mapa Kohonena (Ang. Self Organizing Map, SOM). Sieci Kohonena to rodzaj sieci, w której uczenie odbywa się metodą samoorganizującą typu konkurencyjnego. Metoda ta polega na podawaniu sygnałów na wejścia sieci, a następnie wybraniu w drodze konkurencji zwycięskiego neuronu. Zwycięski neuron najlepiej odpowiada wektorowi wejściowemu. SOM jest techniką wizualizacji danych, która pomaga zrozumieć dane wielowymiarowe poprzez zmniejszenie ich wymiaru do postaci mapy. SOM reprezentuje również koncepcję klastrowania, grupując podobne dane razem. Można więc powiedzieć, że mapa Samoorganizująca się zmniejsza wymiar danych i pokazuje podobieństwa wśród danych.

Poszczególne punkty reprezentuje koordynaty neuronu(tekstu) na mapie SOM. Czerwona kropka odpowiada za teksty popierające szczepienia, a zielona przeciw szczepieniom. Po prawej stronie mapy została umieszczona legenda. Kolor tła pola na mapie pozwala ocenić aktywność neuronów w danym obszarze, a właściwie

odległości pomiędzy neuronami w tym obszarze. Neurony zawierające dane o podobnej charakterystyce zlokalizowane są nieopodal siebie. Zatem im jaśniejszy kolor danego obszaru mapy, tym większe są odległości od neuronów, a więc w tym obszarze zlokalizowane są mało aktywne neurony. Gdy dystans między neuronami jest mały, dany obszar mapy ma ciemniejszy kolor. Są na nim zlokalizowane próbki o podobnej charakterystyce.

Poniżej została przedstawiona mapa uzyskana na podstawie danych zawierających wszystkie istotne słowa. Ważność słowa została określona na podstawie danych z svm.

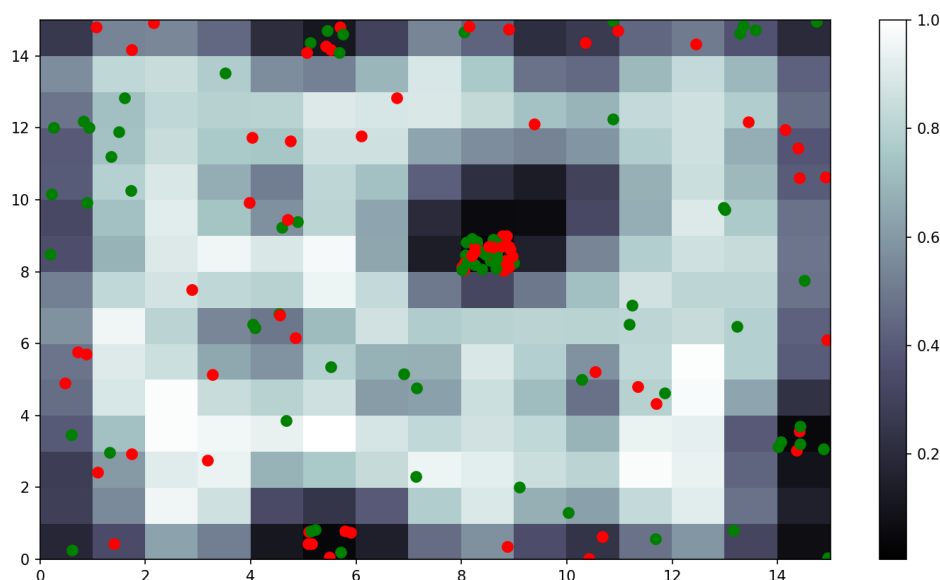


Samoorganizująca się mapa Kohonena. Model stworzony na wszystkich ważnych słowach.

Można zauważyć, że punkty na mapie rozmieszczone są w losowy sposób. Można wyciągnąć wniosek, że SOM trudno wyraźnie sklasyfikować teksty do odpowiednich klas. Taka dystrybucja sugeruje, że teksty obydwu klas charakteryzują się podobnymi cechami. Z drugiej strony, widoczne są jednak obszary, gdzie znajduje się więcej punktów należących do konkretnej klasy. Taka sytuacja wskazuje na to, że niektóre teksty mają w sobie cechy, które umożliwiają stwierdzenie ich podobieństwa. Kolory pól mapy wskazują na to, że neurony w całym jej obszarze były tak samo aktywne podczas procesu uczenia.

4.6. Wpływ ilości istotnych słów mapę Kohonena

W celu dokładniejszej analizy charakterystyk tekstów, została wygenerowana kolejna mapa samoorganizująca. Poniżej została przedstawiona mapa, dla której dane zostały sporządzone podobnie jak w poprzednim przypadku. Tym razem dane wejściowe stanowiło tylko 20 najbardziej znaczących słów, a nie wszystkie. Połowa z wyodrębnionych słów należy do jednej z dwóch klas tekstów.

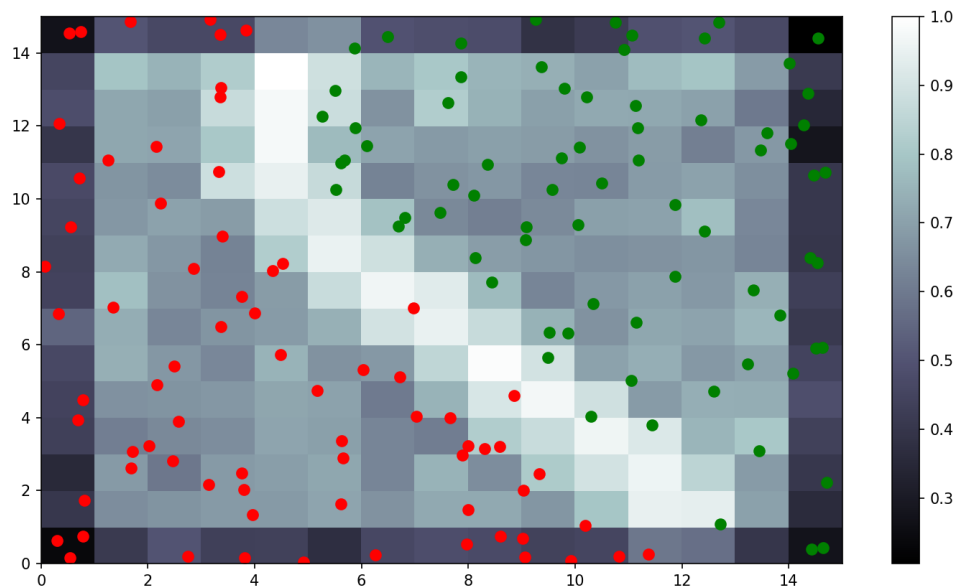


Samoorganizująca się mapa Kohonena. Model stworzony na 20 najważniejszych słowach.

Wyodrębnienie tylko 20 najważniejszych słów znacząco wpłynęło na dystrybucję punktów na mapie. Widoczne są grupy neuronów w kilku obszarach mapy. Skupiska te składają się jednocześnie z próbek obydwu tekstów. Na podstawie ciemnego koloru tych obszarów mapy, można stwierdzić, że to właśnie te miejsca były wyraźnie bardziej aktywne w porównaniu do jasnych obszarów pozostałej części mapy.

4.7. Mapa Kohonena dla danych z kolumną binarną

Następnie została wygenerowana kolejna mapa Kohonena. Do danych dodano dodatkową kolumnę zawierającą wartość binarną, zgodnie z tym czy tekst popiera lub jest przeciwny szczepieniom. Wygenerowana mapa została przedstawiona poniżej.



Samoorganizująca się mapa Kohonena. Dane z dodatkową kolumną.

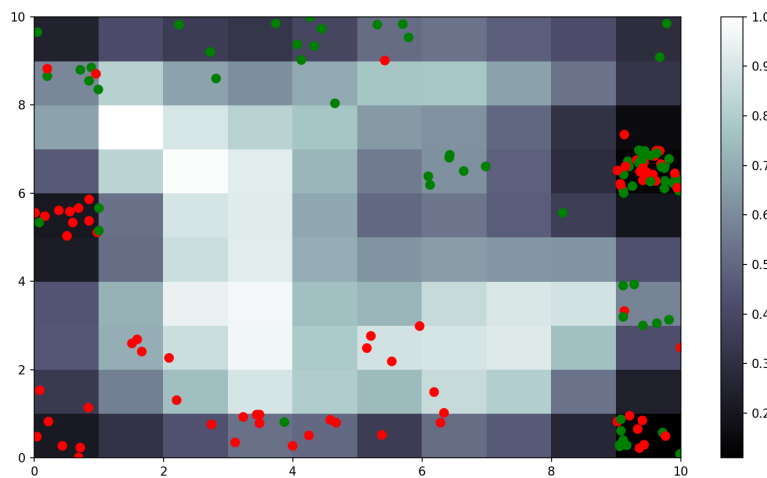
W porównaniu z poprzednią grafiką, widoczna jest znaczna różnica dystrybucji neuronów na mapie. Teksty z danych klas są zlokalizowane w dużych grupach, wyraźnie od siebie oddzielonych. Neurony należące do tej samej klasy zachowują jednak spore odstępstwa od siebie nawzajem. Najrzadziej aktywowane były neurony na granicy obydwu grup - wizualizuje to jasny obszar na mapie przebiegający przez jej środek. Można wyciągnąć wniosek, że stworzenie dodatkowej kolumny w tabeli danych wejściowych znacznie ułatwiło klasyfikację danych na mapie.

4.8. Mapa Kohonena dla danych z funkcją chi2

Następnie została wygenerowana kolejna mapa Kohonena, na której wejście zostały podane dane pochodzą z tekstów, z których usunięto zbędne słowa poprzez wykorzystanie metod zaimplementowanych w bibliotece NLTK (Ang. Natural Language Tagging Toolkit). Sieć wykorzystywała funkcję Chi2, która została zaimplementowana w celu obliczania testu zgodności chi-kwadrat (Ang. Chi-squared test).²⁵ Funkcja Chi2 mierzy zależność między zmiennymi stochastycznymi, więc jej wykorzystanie pozwala usunąć cechy, które najprawdopodobniej są niezależne od klasy, a zatem nieistotne dla klasyfikacji. Funkcja pozwala wybrać tylko najbardziej

²⁵ <https://www.simplilearn.com/tutorials/statistics-tutorial/chi-square-test>

znaczące słowa. Mapa samoorganizująca, na której wejście zostały tak przygotowane dane, została przedstawiona poniżej.

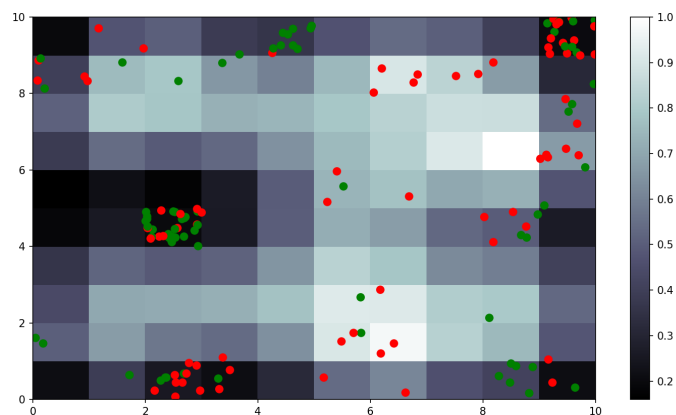


Samoorganizująca się mapa Kohonena. Analizowane z algorytmem Chi2. Zostało wykorzystane tylko 13 najbardziej znaczących słów.

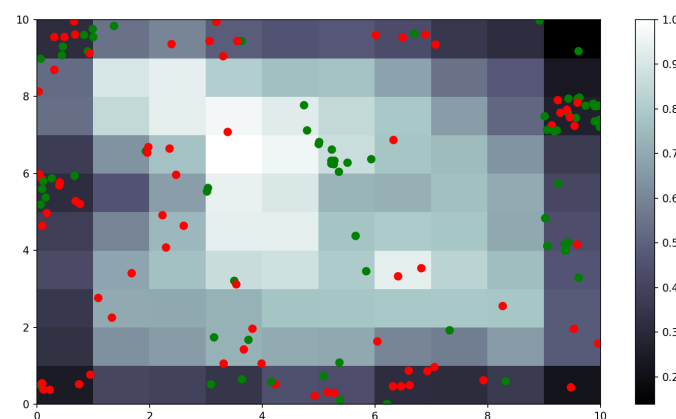
Widoczne są wyraźnie skupiska neuronów w danych obszarach. W kilku przypadkach skupiska składają się z tekstów przypisanych do jednej klasy, co oznacza, że klasyfikator rozpoznał wyraźne cechy podobieństwa między innymi tekstami. Na mapie są też grupy neuronów złożone z tekstów obydwu klas. Takie skupiska charakteryzują się dużą gęstością - punkty są położone bardzo blisko siebie. W tych lokalizacjach neurony były najbardziej aktywne. Najrzadziej pobudzane były neurony w środkowym obszarze mapy - liczba zlokalizowanych tam neuronów jest relatywnie niewielka.

4.9. Wpływ liczby istotnych słów na mapę Kohonena

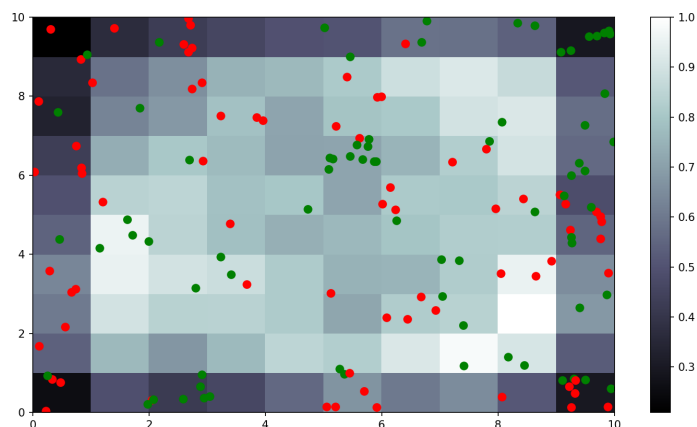
Kolejnym etap było sprawdzenie, w jaki sposób ilość najbardziej istotnych słów wpływa na rozmieszczenie neuronów na mapie. Zostały stworzone trzy listy (tzw. whitelist), zawierające kolno 10, 20 i 30 kluczowych słów. Poniżej zostały przedstawione mapy stworzone na podstawie ich podstawie.



Mapa samoorganizująca na podstawie whitelist o długości 10.



Mapa samoorganizująca na podstawie whitelist o długości 20.



Mapa samoorganizująca na podstawie whitelist o długości 30.

Im więcej słów znajduje się w whitelist, tym większe są odstępstwa pomiędzy punktami na mapie. Na mapie stworzonej na podstawie whitelist o długości 10 widoczne są wyraźne skupiska neuronów należących do obydwu klas. Ten fakt sugeruje, że

klasyfikator nie działał w pełni poprawnie, ponieważ nie przypisywał poprawnie tekstów do ich klas. Na mapie stworzonej na podstawie whitelist o długości 20 widoczne jest więcej skupisk neuronów. Mimo to, mapa ta ma podobną dystrybucję punktów do mapy stworzonej na podstawie whitelist o długości 30. Punkty są rozproszone w miarę losowo na mapie, chociaż widoczne są obszary gdzie zlokalizowane są neurony należące do tej samej klasy. Większa liczba słów w whitelist wpływa pozytywnie na jakość klasyfikacji.

4.10. Wpływ tagowania słów na klasyfikator

W kolejnej części analizy sprawdzono, w jaki sposób zbiór słów wejściowych wpływa na wyniki klasyfikacji. Dzięki wykorzystaniu biblioteki NLTK za pomocą tzw. tagów można określić do jakiej części mowy należy dane słowo.²⁶ Początkowo w zbiorze słów zostały tylko rzeczowniki przymiotniki i przysłówki. Raporty klasyfikacji klasyfikatora Keras oraz SVM zostały już przeanalizowane w tym sprawozdaniu. Poniżej przedstawiono je ponownie w celu łatwiejszego porównania.

----- Evaluation on Test Data -----				
	precision	recall	f1-score	support
False	0.68	0.62	0.65	21
True	0.56	0.62	0.59	16
accuracy			0.62	37
macro avg	0.62	0.62	0.62	37
weighted avg	0.63	0.62	0.62	37

Raport klasyfikacji dla klasyfikatora Keras trenowanego na danych zawierających rzeczowniki, przymiotniki i przysłówki

²⁶ <https://cs.nyu.edu/~grishman/jet/guide/PennPOS.html>

```

----- Evaluation on Test Data SVM -----
[[14  4]
 [ 7 12]]

```

	precision	recall	f1-score	support
False	0.67	0.78	0.72	18
True	0.75	0.63	0.69	19
accuracy			0.70	37
macro avg	0.71	0.70	0.70	37
weighted avg	0.71	0.70	0.70	37

Raport klasyfikacji dla klasyfikatora SVM trenowanego na danych zawierających rzeczowniki, przymiotniki i przysłówki

Następnie ze zbioru słów zostały usunięte wszystkie przysłówki. Raport klasyfikacji obydwu klasyfikatorów bazującego tylko na rzeczownikach i przymiotnikach został przedstawiony poniżej.

```

----- Evaluation on Test Data -----

```

	precision	recall	f1-score	support
False	0.72	0.93	0.81	14
True	0.95	0.78	0.86	23
accuracy			0.84	37
macro avg	0.83	0.86	0.83	37
weighted avg	0.86	0.84	0.84	37

Raport klasyfikacji dla klasyfikatora Keras trenowanego na danych zawierających rzeczowniki i przymiotniki

```

----- Evaluation on Test Data SVM -----
[[11  4]
 [13  9]]

```

	precision	recall	f1-score	support
False	0.46	0.73	0.56	15
True	0.69	0.41	0.51	22
accuracy			0.54	37
macro avg	0.58	0.57	0.54	37
weighted avg	0.60	0.54	0.53	37

Raport klasyfikacji dla klasyfikatora SVM trenowanego na danych zawierających rzeczowniki i przymiotniki

W celu sprawdzenia wpływu słów danej części mowy na charakterystykę klasyfikatora, najpierw został sprawdzony wpływ usunięcia wszystkich przymiotników ze zbioru słów. Poniżej został przedstawiony raport klasyfikacji klasyfikatora Keras oraz SVM bazujących tylko na rzeczownikach oraz przysłówkach.

```
----- Evaluation on Test Data -----
```

	precision	recall	f1-score	support
False	0.61	0.69	0.65	16
True	0.74	0.67	0.70	21
accuracy			0.68	37
macro avg	0.67	0.68	0.67	37
weighted avg	0.68	0.68	0.68	37

Raport klasyfikacji dla klasyfikatora Keras trenowanego na danych zawierających rzeczowniki oraz przysłówki

```
----- Evaluation on Test Data SVM -----
```

```
[[12  2]
 [11 12]]
```

	precision	recall	f1-score	support
False	0.52	0.86	0.65	14
True	0.86	0.52	0.65	23
accuracy			0.65	37
macro avg	0.69	0.69	0.65	37
weighted avg	0.73	0.65	0.65	37

Raport klasyfikacji dla klasyfikatora SVM trenowanego na danych zawierających rzeczowniki oraz przysłówki

Następnie ze zbioru słów zostały usunięte wszystkie przysłówki i przymiotniki. Raport klasyfikacji klasyfikatorów bazującego tylko na rzeczownikach został przedstawiony poniżej.

```

----- Evaluation on Test Data -----

```

	precision	recall	f1-score	support
False	0.55	0.79	0.65	14
True	0.82	0.61	0.70	23
accuracy			0.68	37
macro avg	0.69	0.70	0.67	37
weighted avg	0.72	0.68	0.68	37

Raport klasyfikacji dla klasyfikatora Keras trenowanego na danych zawierających rzeczowniki

```

----- Evaluation on Test Data SVM -----
[[14  5]
 [ 5 13]]

```

	precision	recall	f1-score	support
False	0.74	0.74	0.74	19
True	0.72	0.72	0.72	18
accuracy			0.73	37
macro avg	0.73	0.73	0.73	37
weighted avg	0.73	0.73	0.73	37

Raport klasyfikacji dla klasyfikatora SVM trenowanego na danych zawierających rzeczowniki

Jedną z kluczowych cech klasyfikatora jest jego dokładność. W celu wygodniejszego porównania dokładności klasyfikatorów w konkretnych przypadkach, została sporządzona poniższa tabela.

	Części mowy zawarte w zbiorze analizowanych słów			
	rzeczowniki, przymiotniki, przysłówki	rzeczowniki, przymiotniki	rzeczowniki, przysłówki	rzeczowniki
Dokładność klasyfikatora Keras dla danych treningowych	0,62	0,84	0,68	0,68
Dokładność klasyfikatora SVM dla danych treningowych	0,70	0,54	0,65	0,73

Klasyfikator Keras działa najlepiej dla danych zawierających tylko rzeczowniki i przymiotniki - jego dokładność wynosi 84%. W pozostałych przypadkach osiąga około 20% niższą dokładność. Taki rezultat sugeruje wniosek, że dla tego klasyfikatora przysłówki ograniczają jego dokładność.

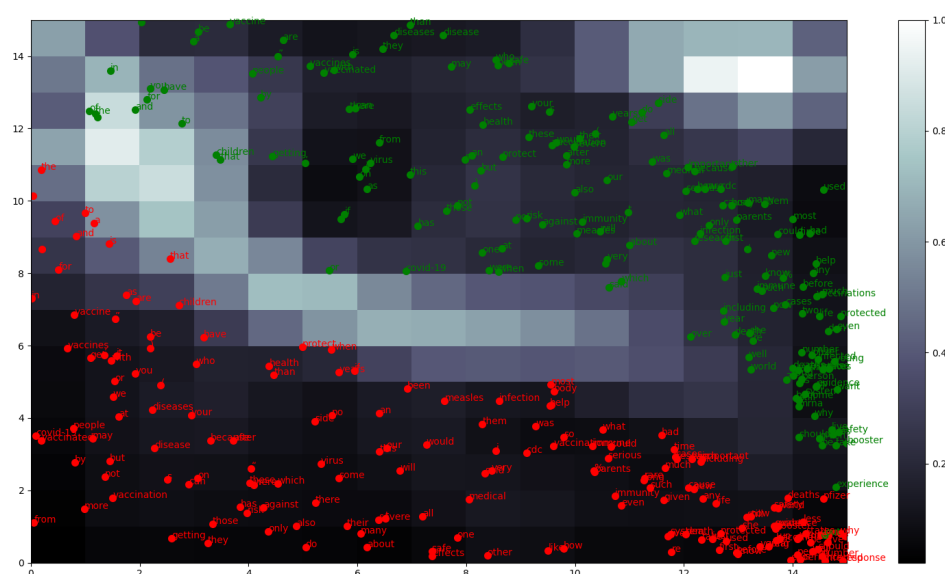
Klasyfikator SVM działa najlepiej dla danych zawierających tylko rzeczowniki - osiągnął dokładność równą 73%. Zróżnicowanie dokładności jest o wiele większe niż w przypadku pierwszego klasyfikatora. Najgorzej poradził sobie w przypadku gdy brane były pod uwagę tylko rzeczowniki i przymiotniki - dokładność wynosiła 54%.

Na podstawie wyników zamieszczonych w tabeli można wywnioskować, że korzystając z klasyfikatora Keras najlepiej brać pod uwagę tylko rzeczowniki i przymiotniki. Natomiast dla takich danych klasyfikator SVM osiągnął niesatysfakcjonujące rezultaty. Dla tego klasyfikatora najlepiej stosować dane zawierające tylko rzeczowniki.

4.11. Badanie siatki pojęciowej

W celu badania siatki pojęciowej została wygenerowana macierz podobieństw między najważniejszymi słowami. Istotność słów określona jest przez wartość Permutation Importance. Macierz została stworzona osobno dla tekstów za szczepieniem oraz przeciw szczepieniom. Podobieństwo słów zostało zmierzone za

pomocą funkcji zaimplementowanych w bibliotece Word2Vec.²⁷ Dwie powstałe macierze łączone są w jedną a następnie rezultat podawany jest na wejście mapy SOM. Są to wektory kontekstu wygenerowane przez tę bibliotekę dla słów z obydwu klas. Poniżej została przedstawiona stworzona mapa SOM. Każdy z neuronów na mapie został opisany słowem, który jest z nim związany. Mapa ma na celu pokazać podobieństwo danych słów. Pojedyncze słowa pojawiają się jako podpisy neuronów w obydwu grupach, ponieważ mogą być one istotne dla każdej z klas.



Mapa SOM pokazująca podobieństwo słów w każdej z klas.

Dzieląc zbiór danych wejściowych na dwie części - na przykład 80% oraz 20%, można zbadać wpływ douczania sieci neuronowej na dystrybucję neuronów na mapie SOM. Mapa będzie wygenerowana odpowiednio dla podstawowej sieci neuronowej oraz douczanej sieci neuronowej. Pozwala to badać zmiany w średnich odległościach pomiędzy neuronami pod wpływem nowych danych treningowych. Dzięki temu można ocenić jak bardzo zmienia się podobieństwo danych słów.

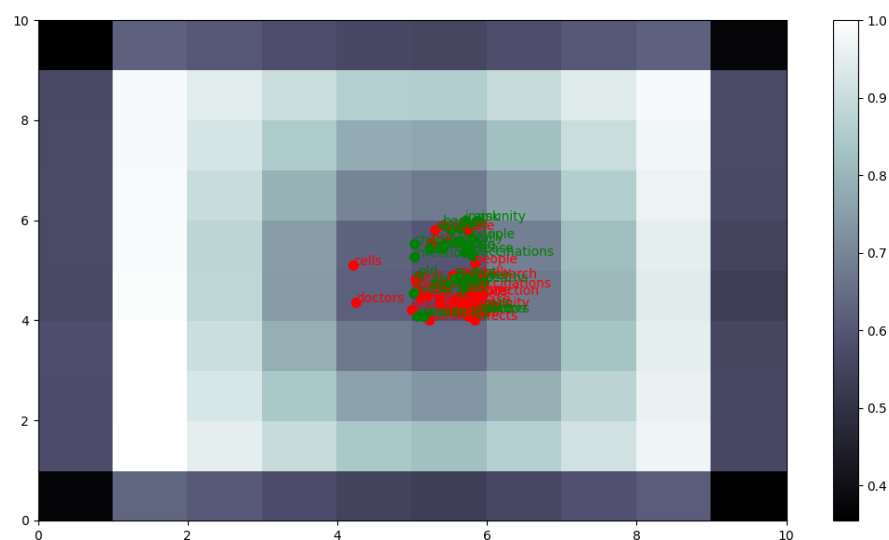
Siatka pojęciowa nauczana na 80% danych zostanie douczona poprzez wykorzystanie 20% dodatkowych danych - najpierw popierających szczepienie, a następnie przeciwnym szczepieniom. Badamy jak ta metoda wpływa na rozmieszczenie słów na mapie SOM.

²⁷ <https://www.geeksforgeeks.org/python-word-embedding-using-word2vec/>

Co więcej, tempo nauki sieci zależy od parametru tempa uczenia (Ang. Learning Rate) mapy SOM.²⁸ Learning rate odpowiada za aktualizację wag w neuronach sieci. Wartość ta stopniowo maleje z biegiem czasu.

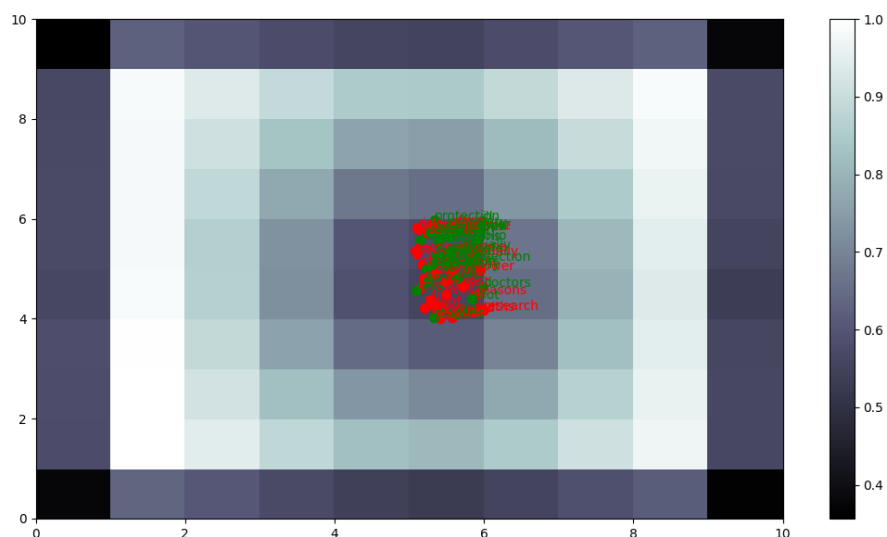
Analiza odległości pomiędzy słowami pozwala uzyskać informacje o zagęszczeniu oraz rozproszeniu słów. Zostanie obliczona suma odległości pomiędzy neuronami w poszczególnych klasach. Obliczanie sumy odległości będzie wykonywane przed oraz po douczeniu. Zostanie także obliczona wartość średnia odległości wykorzystując liczbę słów należących do danej klasy.

Na początku Learning rate został ustawiony jako wartość 0,0001.



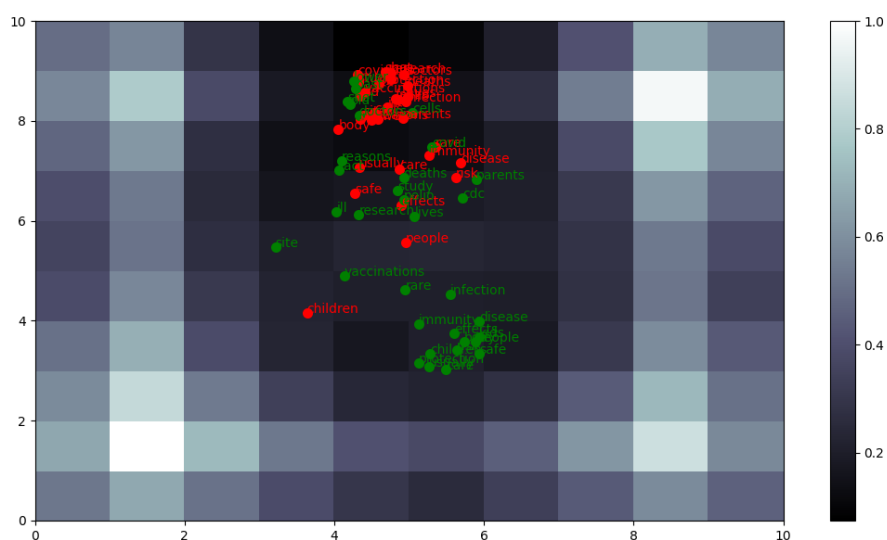
*Mapa SOM przedstawiająca podobieństwo słów dla 80% wszystkich danych.
Learning Rate wynosi 0,0001.*

²⁸ <https://medium.com/machine-learning-researcher/self-organizing-map-som-c296561e2117>

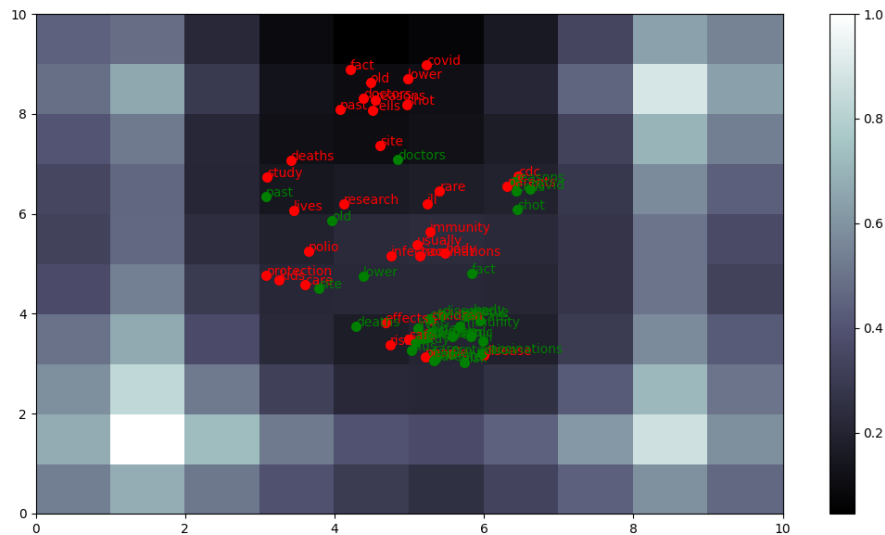


Mapa SOM przedstawiająca podobieństwo słów dotrenowana 20% wszystkich danych. Learning Rate wynosi 0,0001.

Następnie Learning rate został ustawiony jako wartość 0,001.

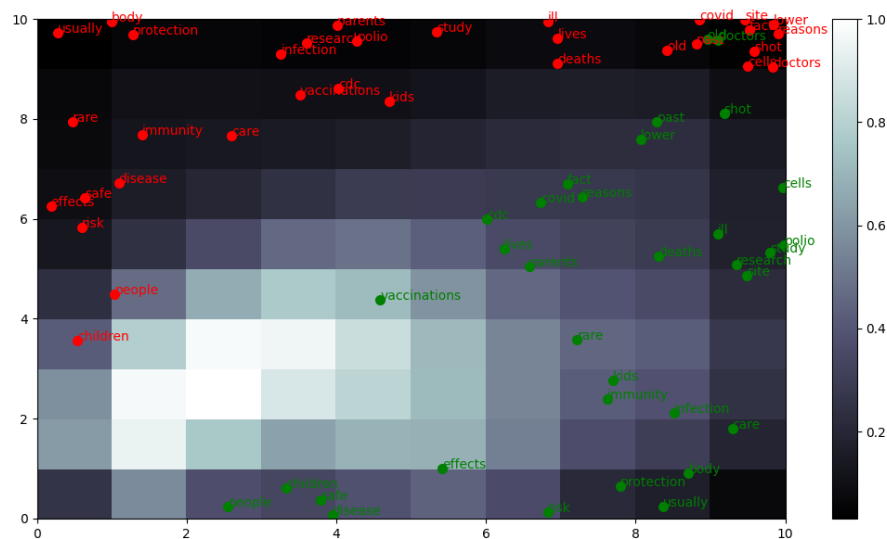


Mapa SOM przedstawiająca podobieństwo słów dla 80% wszystkich danych. Learning Rate wynosi 0,001.

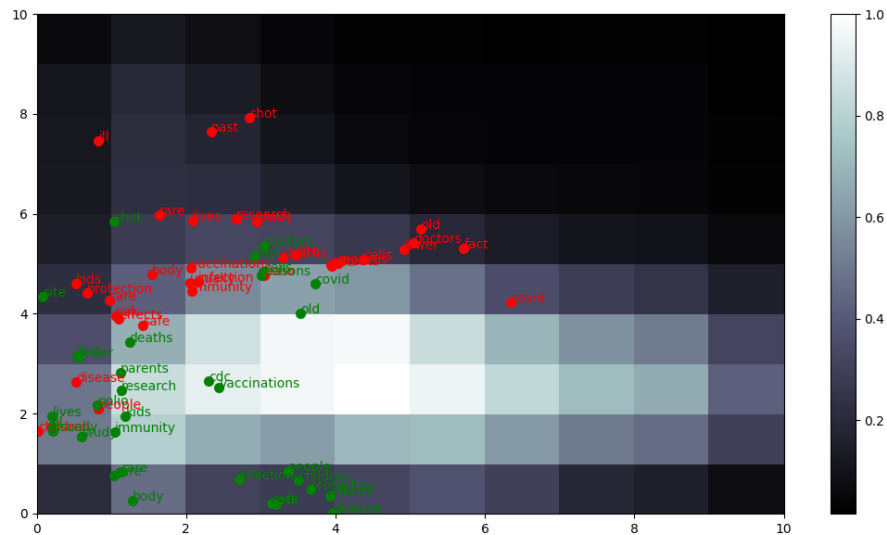


Mapa SOM przedstawiająca podobieństwo słów dotrenowana 20% wszystkich danych. Learning Rate wynosi 0,001.

Następnie Learning rate został ustawiony jako wartość 0,01.

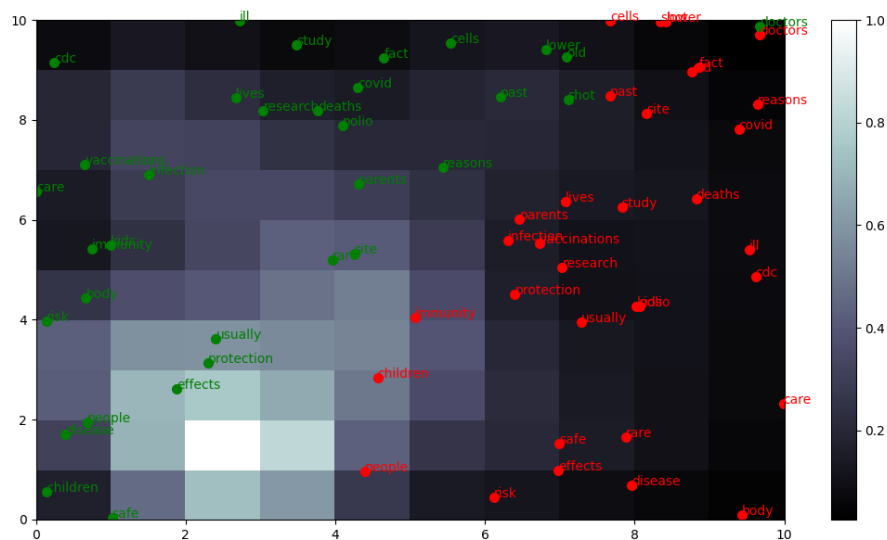


Mapa SOM przedstawiająca podobieństwo słów dla 80% wszystkich danych. Learning Rate wynosi 0,01.

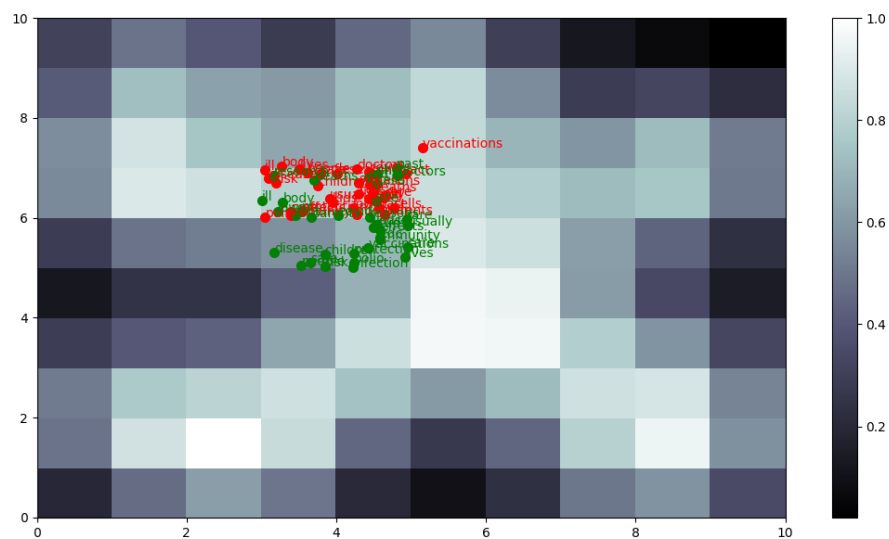


Mapa SOM przedstawiająca podobieństwo słów dotrenowana 20% wszystkich danych. Learning Rate wynosi 0,01.

Następnie Learning rate został ustawiony jako wartość 0,1.

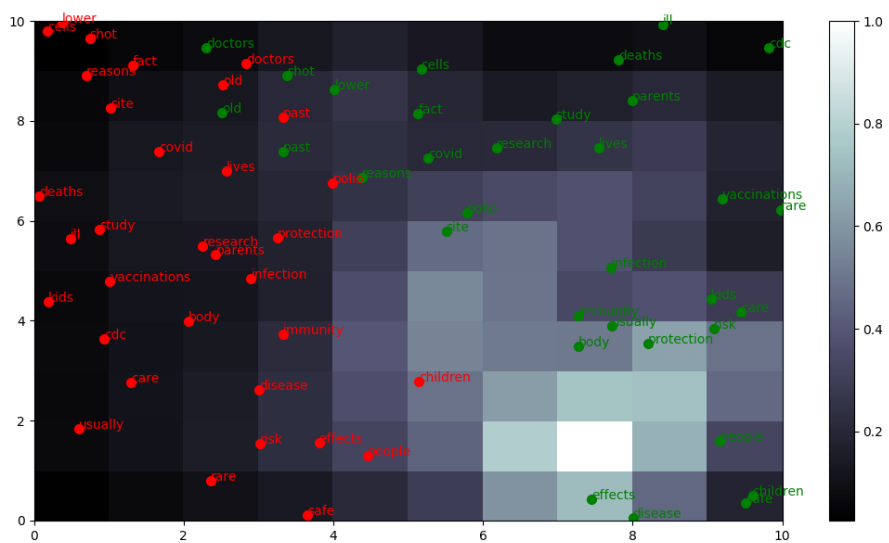


Mapa SOM przedstawiająca podobieństwo słów dla 80% wszystkich danych. Learning Rate wynosi 0,1.

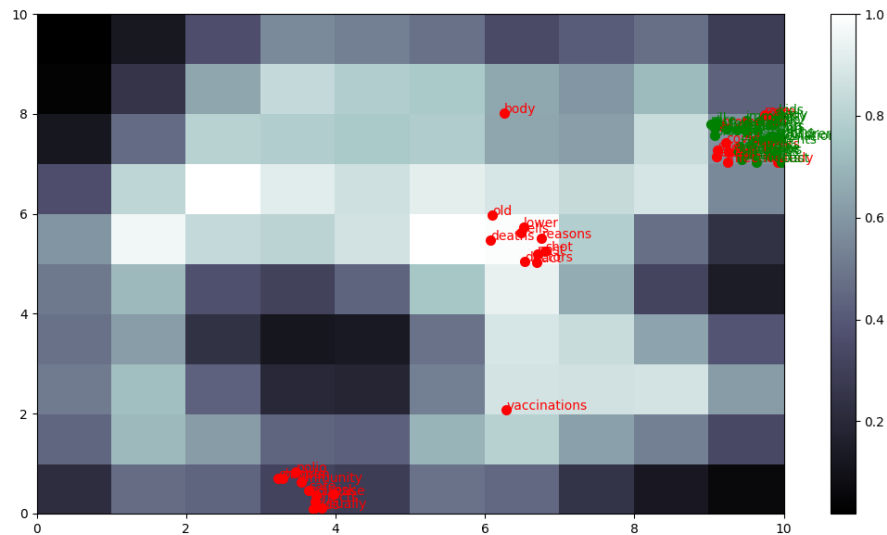


Mapa SOM przedstawiająca podobieństwo słów dotrenowana 20% wszystkich danych. Learning Rate wynosi 0,1.

Następnie Learning rate został ustawiony jako wartość 0,15.



Mapa SOM przedstawiająca podobieństwo słów dla 80% wszystkich danych. Learning Rate wynosi 0,15.



Mapa SOM przedstawiająca podobieństwo słów dotrenowana 20% wszystkich danych. Learning Rate wynosi 0,15.

Analiza odległości pomiędzy neuronami w danych przypadkach prezentuje się następująco:

DataType	LearningRate	sumFor	avgFor	sumAgainst	avgAgainst
Before	0,15	4254,42	128,92	4898,61	148,44
After	0,15	4704,82	142,57	531,65	16,11
Difference	0,15	450,4	13,65	-4366,96	-132,33
Before	0,1	4493,85	136,18	4996,53	151,41
After	0,1	921,47	27,92	1171,22	35,49
Difference	0,1	-3572,38	-108,26	-3825,31	-115,92
Before	0,01	5065,24	153,49	4682,85	141,9
After	0,01	2787,75	84,48	2899,46	87,86
Difference	0,01	-2277,49	-69,01	-1783,39	-54,04
Before	0,001	1373,73	41,63	2634,37	79,83
After	0,001	2674,61	81,05	1766,54	53,53
Difference	0,001	1300,88	39,42	-867,83	-26,3
Before	0,0001	808,88	24,51	887,59	26,9
After	0,0001	930,39	28,19	740,04	22,43
Difference	0,0001	121,51	3,68	-147,55	-4,47