.

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ INŻYNIERII METALI I INFORMATYKI PRZEMYSŁOWEJ

Logika rozmyta jako forma reprezentacji wiedzy w systemach sterowania lub podejmowania decyzji.

Zastosowanie sterownika logicznego w analizie zachowań użytkowników na przykładzie portali społecznościowych.

Autor: *Maciej Adam Krzywda*
Kierunek studiów: Inżynieria Obliczeniowa
Nr albumu: *293102*

Kraków, 2020

.

1. Opis problemu, który ma być rozwiązany z wykorzystaniem systemowej reprezentacji wiedzy z wykorzystaniem logiki rozmytej

Obecnie prawie każdy z nas posiada konto na profilu społecznościowym takim jak Facebook, Twitter czy Instagram. Prowadzimy tam bardzo szeroko pojętą działalność - wrzucamy zdjęcia, wyrażamy opinie, prowadzimy dyskusje publiczne oraz prywatne. Wraz z rozwojem i popularnością tychże portali spotykamy się z narastająca ilością niechcianych wiadomości, oraz podejrzanych profili użytkowników. Najczęściej taka osoba to troll lub spamer, ale równie często jest to oszust który w łatwy sposób może nas zaatakować. Wychodząc naprzeciw postanowiłem przygotować prototyp sterownika rozmytego do analizy czy dany profil jest tzw. profilem uciążliwym czyli spamerem. Korzystając z własnego doświadczenia oraz mechanizmów działania różnych portali społecznościowych postanowiłem przyjrzeć się Twitterowi pod względem zachowań użytkowników.

Użytkownika można scharakteryzować korzystając z 6 zmiennych (w modelu uproszczonym) mianowicie: Tweets, Tweets & repiles, Media, Likes, Join Date oraz parametry dotyczące publikowanych postów czyli CountOfTags, CountOfChars, CountOfWords and CountOfURLs w jednym Tweecie.

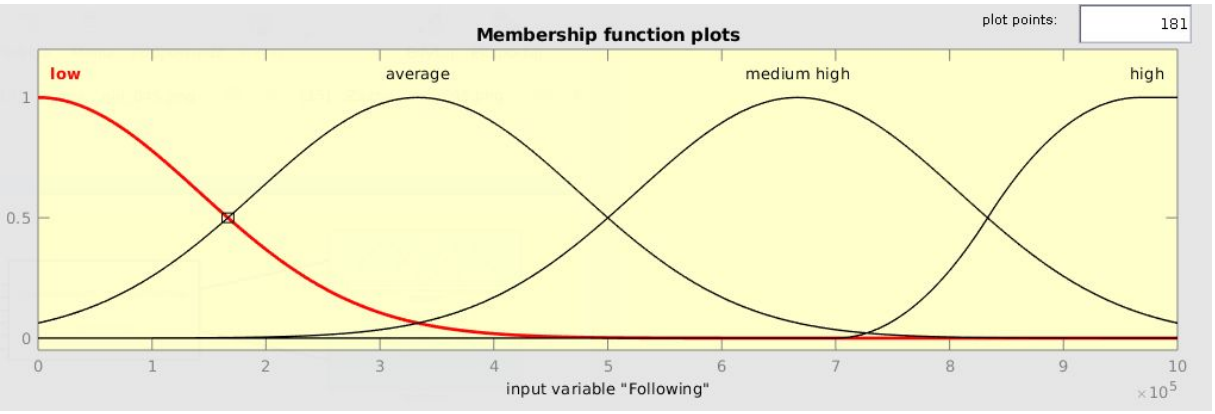
Obecnie można publikować posty o długości max 140 znaków, co mocno wpłynęło na zakresy zmiennych o których więce będzie mowa w następnym punkcie.

.

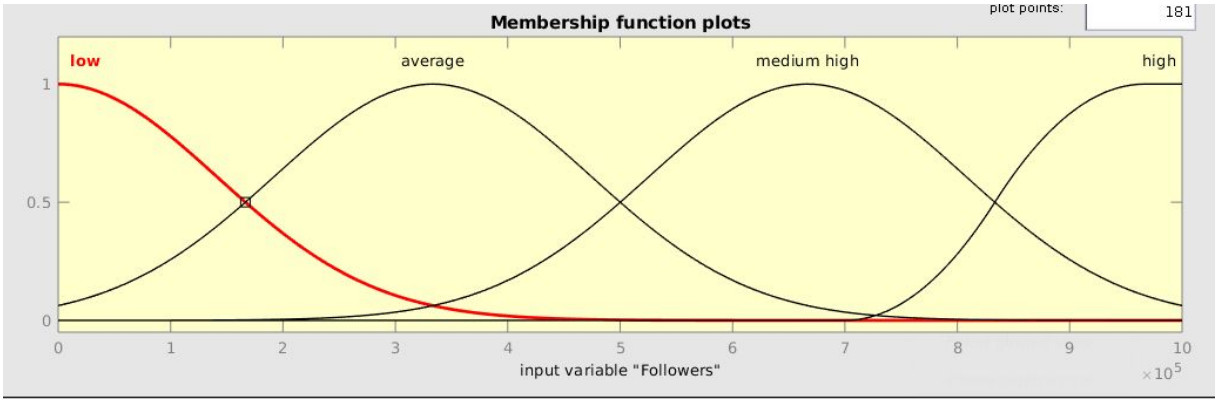
2. Reprezentacja wybranych zmiennych lingwistycznych w postaci zbiorów rozmytych

W swoim sterowniku rozmytym wykorzystałem 7 wejść oraz 2 wyjścia. Zmienne lingwistyczne reprezentujące sygnały wejściowe reprezentowane są przez parametry które możemy określić osobę publikująca post. Natomiast zmienne wyjściowe reprezentują wartośc określająca czy dany wpis jest spamem czy nie.

Zastosowanie zmiennych gaussowskich które dominują w reprezentacji funkcji przynależności ma swoje odzwierciedlenie w stanie faktycznym jaki ma miejsce na portalach społecznościowych. Mianowicie wzrost albo spadek ilości followersow nie powoduje diametralnej zmiany i nie zaburza całego procesu wnioskowania, co innego w przypadku sygnałów na wyjściu - tam konieczne było ukazanie ostrości, która mocno powiązana jest z realnymi uczuciami ludzkimi dot. spamu.

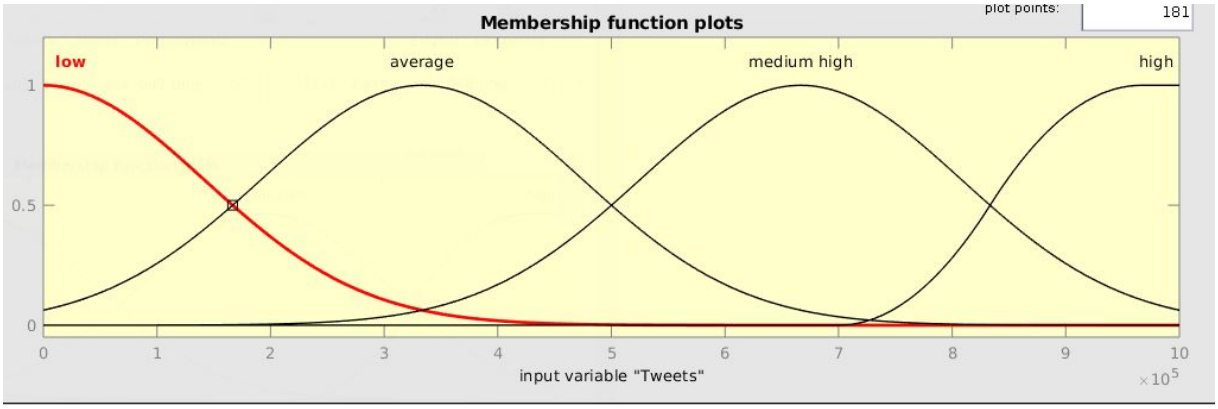


Name='Following'
Range=[0 1000000]
MF1='low':'gaussmf',[141500 0]
MF2='average':'gaussmf',[141600 333300]
MF3='medium high':'gaussmf',[141500 666700]
MF4='high':'pimf',[700100 966700 1100000 1633000]

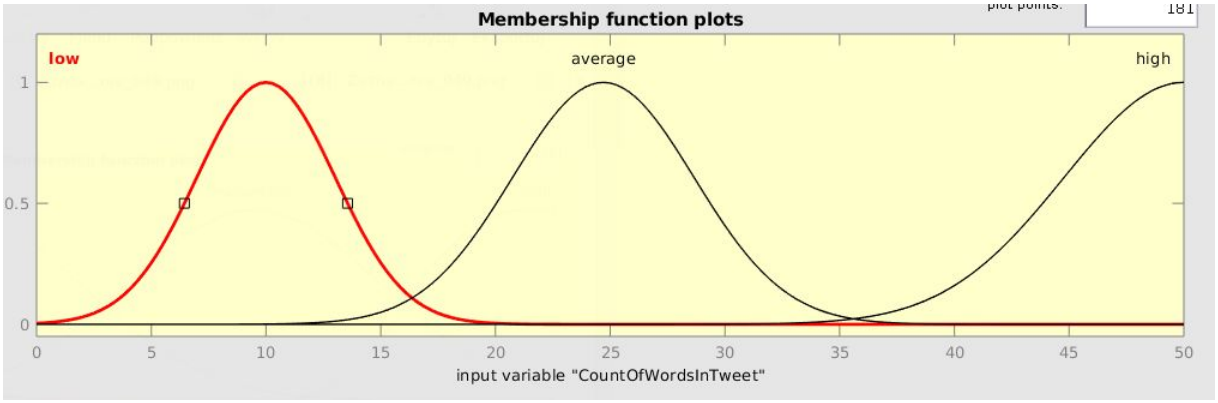


Name='Followers'
Range=[0 1000000]
MF1='low':'gaussmf',[141500 0]
MF2='average':'gaussmf',[141600 333300]
MF3='medium high':'gaussmf',[141500 666700]
MF4='high':'pimf',[700100 966700 1100000 1633000]

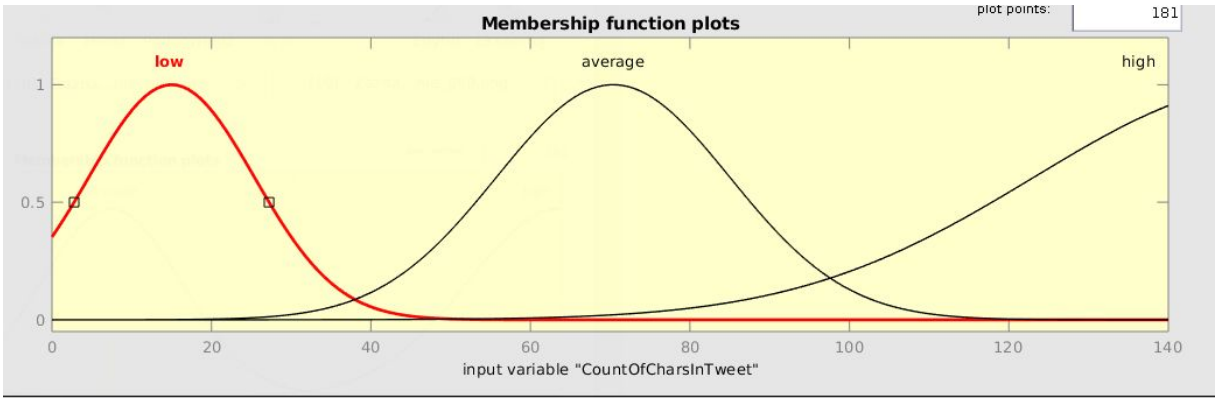
.



Name='Tweets'
Range=[0 1000000]
MF1='low':'gaussmf',[141500 0]
MF2='average':'gaussmf',[141600 333300]
MF3='medium high':'gaussmf',[141500 666700]
MF4='high':'pimf',[700100 966700 1100000 1633000]

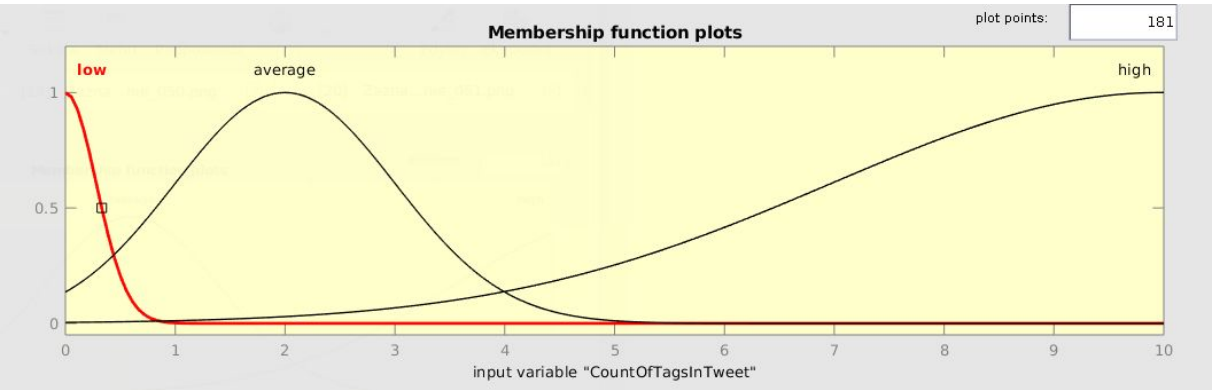


Name='CountOfWordsInTweet'
Range=[0 50]
MF1='low':'gaussmf',[3.02 10]
MF2='average':'gaussmf',[3.95810921531052 24.7]
MF3='high':'gaussmf',[5.22400313669219 50]

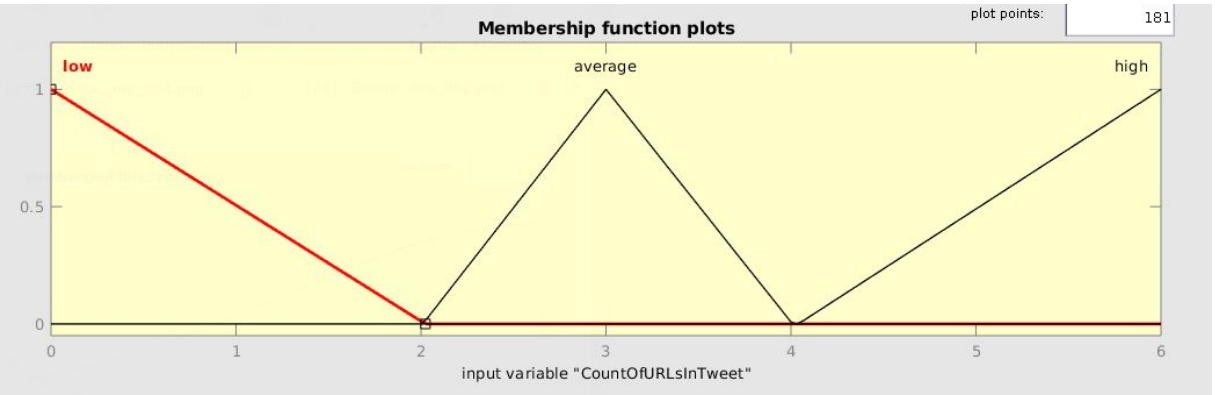


Name='CountOfCharsInTweet'
Range=[0 140]
MF1='low':'gaussmf',[10.380599781298 15]
MF2='average':'gaussmf',[14.63 70.3703703703704]
MF3='high':'gaussmf',[29.7 152.851851851852]

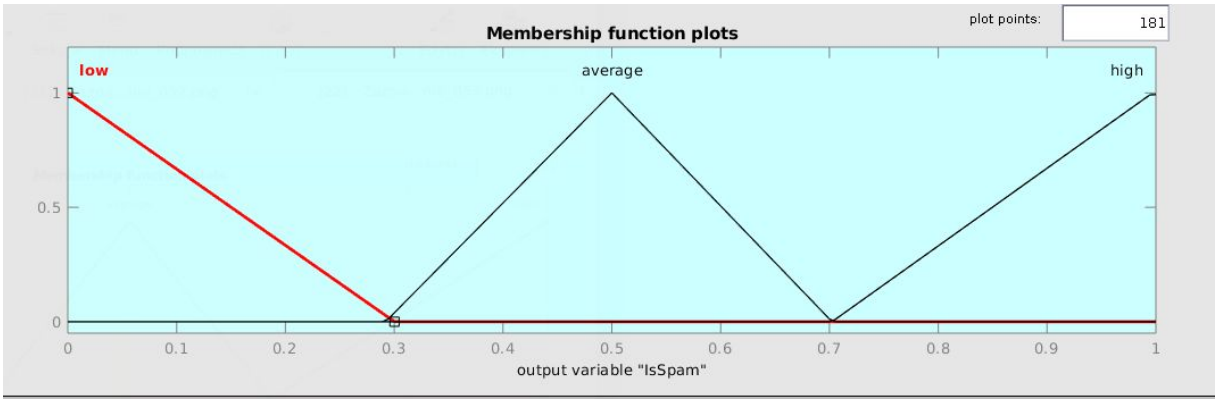
.



Name='CountOfTagsInTweet'
Range=[0 10]
MF1='low':'gaussmf',[0.2809 -5.55e-17]
MF2='average':'gaussmf',[1 2]
MF3='high':'gaussmf',[3 9.974]

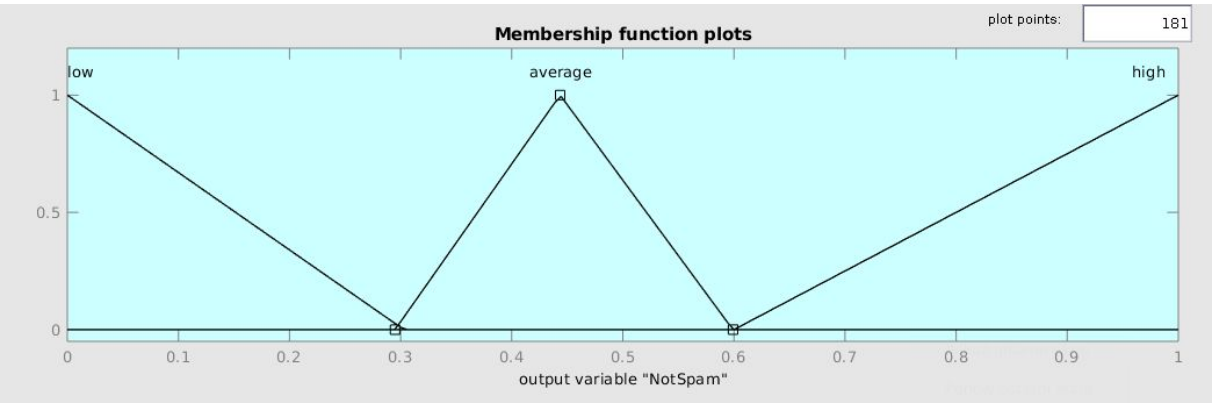


Name='CountOfURLsInTweet'
Range=[0 6]
MF1='low':'trimf',[-2.4 0 2.02380952380952]
MF2='average':'trimf',[2.01 3 4.00793650793651]
MF3='high':'trimf',[4.03968253968254 6 8.4]



Name='IsSpam'
Range=[0 1]
MF1='low':'trimf',[-0.4 0 0.30026455026455]
MF2='average':'trimf',[0.292 0.5 0.702380952380952]
MF3='high':'trimf',[0.702380952380952 0.997 1.4]

.



Name='NotSpam'
Range=[0 1]
MF1='low':'trimf',[-0.4 0 0.302910052910053]
MF2='average':'trimf',[0.295 0.4436 0.599]
MF3='high':'trimf',[0.6 1 1.4]

3. Rozmyta baza wiedzy

Przygotowując bazę wiedzy oparłem się o kilka źródeł. Przeglądałem pozyskane dane z Twitter API, obserwowałem popularne tweety oraz patrząc na profile osób które udzielają się w tzw. gorących tematach. Dodatkowo skontaktowałem się ze znajomymi, w celu weryfikacji moim reguł oraz możliwości dodawania swoich.

1. If (Following is low) and (Followers is high) then (IsSpam is high)(NotSpam is low)
2. If (Following is average) and (Followers is medium high) then (IsSpam is average)(NotSpam is average)
3. If (Following is medium high) and (Followers is average) then (IsSpam is low)(NotSpam is high)
4. If (Following is high) and (Followers is low) then (IsSpam is low)(NotSpam is high)
5. If (Tweets is low) then (IsSpam is low)(NotSpam is high)
6. If (Tweets is average) then (IsSpam is average)(NotSpam is average)
7. If (Tweets is medium high) then (IsSpam is high)(NotSpam is low)
8. If (CountOfWordsInTweet is low) and (CountOfTagsInTweet is high) then (IsSpam is high)(NotSpam is low)
9. If (CountOfWordsInTweet is average) and (CountOfTagsInTweet is average) then (IsSpam is average)(NotSpam is average)
10. If (CountOfWordsInTweet is high) and (CountOfTagsInTweet is low) then (IsSpam is low)(NotSpam is high)
11. If (CountOfCharsInTweet is low) and (CountOfTagsInTweet is high) and (CountOfURLsInTweet is high) then (IsSpam is high)(NotSpam is low)
12. If (CountOfCharsInTweet is average) and (CountOfTagsInTweet is high) and (CountOfURLsInTweet is high) then (IsSpam is average)(NotSpam is average)
13. If (CountOfCharsInTweet is high) and (CountOfTagsInTweet is average) and (CountOfURLsInTweet is average) then (IsSpam is average)(NotSpam is average)
14. If (CountOfCharsInTweet is high) and (CountOfTagsInTweet is low) and (CountOfURLsInTweet is low) then (IsSpam is low)(NotSpam is high)
15. If (CountOfTagsInTweet is high) and (CountOfURLsInTweet is high) then (IsSpam is high)(NotSpam is low)
16. If (CountOfTagsInTweet is average) and (CountOfURLsInTweet is average) then (IsSpam is average)(NotSpam is average)
17. If (CountOfTagsInTweet is low) and (CountOfURLsInTweet is low) then (IsSpam is low)(NotSpam is high)
18. If (CountOfTagsInTweet is high) then (IsSpam is high)(NotSpam is low)
19. If (CountOfURLsInTweet is high) then (IsSpam is high)(NotSpam is low)
20. If (CountOfWordsInTweet is low) and (CountOfCharsInTweet is high) then (IsSpam is high)(NotSpam is low)

.

4. Operatory logiczne

And metod	PROD
Or metod	Nie dotyczy
Implication	MIN
Aggregation	MAX
Defuzzification	BISECTOR

And metod - Każdy z czynników wpływa na ryzyko w innym stopniu. Aby móc wziąć wszystkie te współczynniki pod uwagę zastosowano iloczyn product.
Wzór: $\mu_{followers} \times \mu_{following} \times \mu_{tweets} \times \mu_{CountOfWordssInTweet'} \times \mu_{CountOfCharssInTweet'} \times \mu_{CountOfTagsInTweet} \times \mu_{CountOfURLsInTweet'}$

Implication - Każdy z czynników wpływa na ryzyko w innym stopniu. Aby móc wziąć wszystkie te współczynniki pod uwagę zastosowano implikację Larsena.
Wzór: $\mu_{followers} \times \mu_{following} \times \mu_{tweets} \times \mu_{CountOfWordssInTweet'} \times \mu_{CountOfCharssInTweet'} \times \mu_{CountOfTagsInTweet} \times \mu_{CountOfURLsInTweet'}$

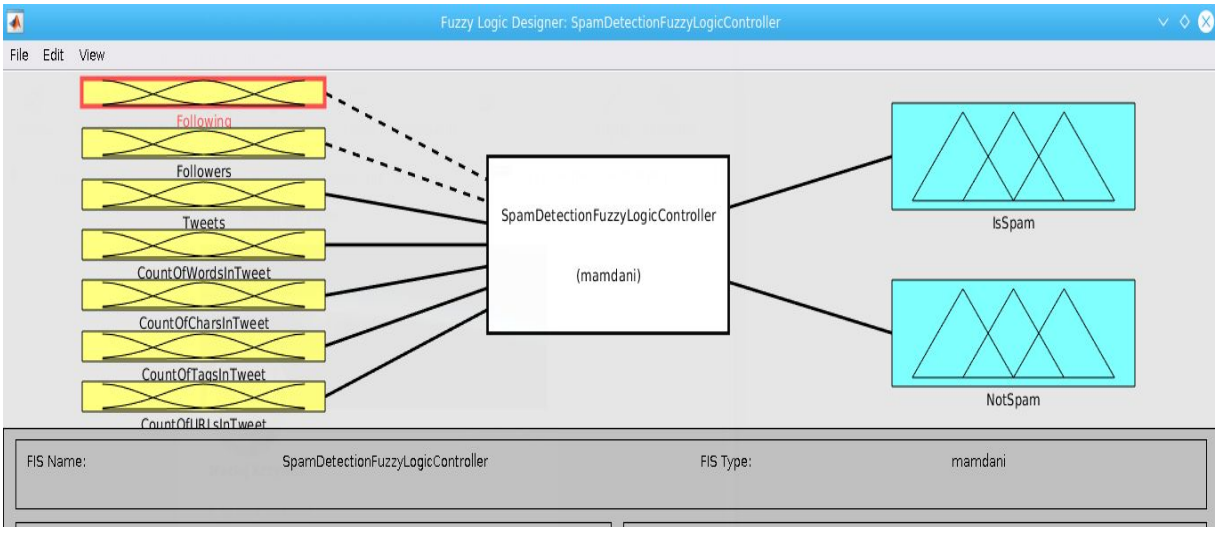
Aggregation - Pod uwagę brana będzie ta przesłanka, która najsilniej wskazuje na wystąpienie danego stopnia ryzyka
Wzór: $\mu_c=[\mu_A, \mu_B]$

Defuzzyfication - W tym systemie chcemy otrzymać wynik wynikający z najsilniej zaktywizowanej przesłanki.

$$\int_{\alpha}^{x^*} \mu_A(x)\ dx = \int_{x^*}^{\beta} \mu_A(x)\, dx \text{ , where } \alpha = \min\{x \mid x \in X\} \text{ and } \beta = \max\{x \mid x \in X\}$$

5. Implementacja

W swojej pracy zastosowałem Fuzzy Logic Toolbox znajdujący się w pakiecie obliczeniowym MATLAB firmy MathWorks.



Jest to pakiet który rozszerza środowisko MATLAB o narzędzia do projektowania systemów opartych na logice rozmytej. Graficzne interfejsu użytkownika (GUI) prowadzą poprzez kolejne kroki systemu projektowania wnioskowania rozmytego. Moduł zapewnia funkcje dla wielu typowych metod logiki rozmytej.

Zawiera między innymi:

- Graficzny interfejs użytkownika (GUI) do budowy rozmytych systemów wnioskujących oraz podglądu i analizy wyników.

.

- Funkcje przynależności do tworzenia systemów rozmytego wnioskowania.
- Obsługa logiki AND, OR i NOT w zdefiniowany przez użytkownika sposób.
- Standardowy system wnioskowania rozmytego typu Mamdani i Sugeno.

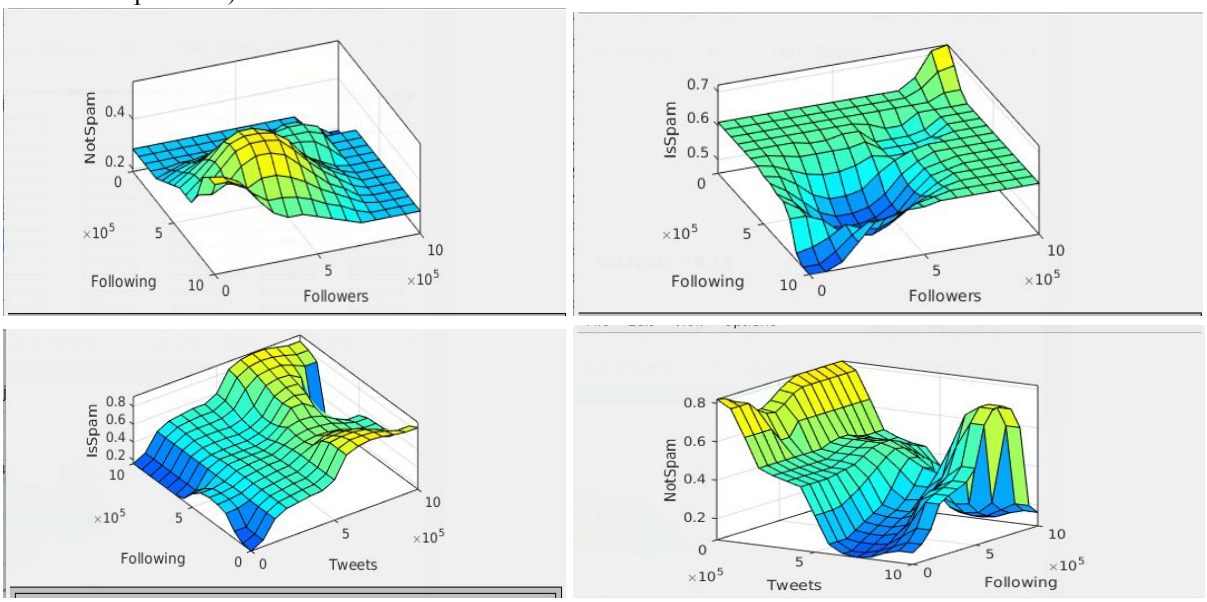
Co więcej jest również możliwość umieszczania systemu rozmytego wnioskowania w modelu Simulinka i późniejsze wygenerowanie go jako kod języka C.

6. Wnioskowanie rozmyte – opis scenariuszy wnioskowania

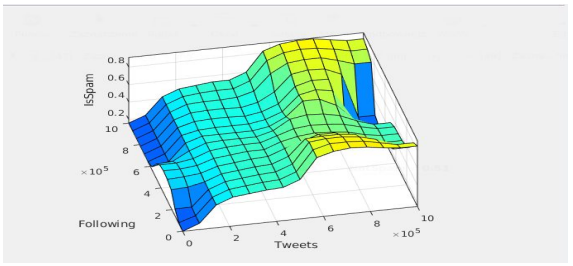
Aby sprawdzić działanie systemu przyjęto 3 scenariusze testowe.

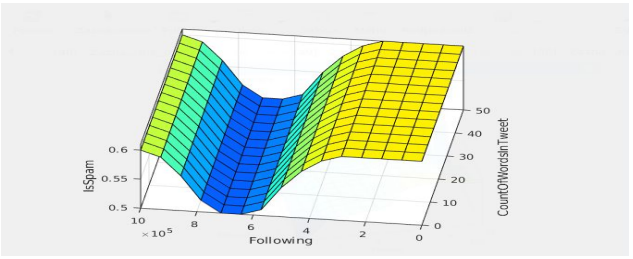
Lp.	Following	Followers	Tweets	CountOfWords	CountOfChars	CountOfTag	CountOfUrls
1	21	14000	1000	20	140	2	0
2	1234	4321	12412	11	140	1	3
3	0	192414	9999	15	140	4	1
4	11	0	107621	39	140	0	0

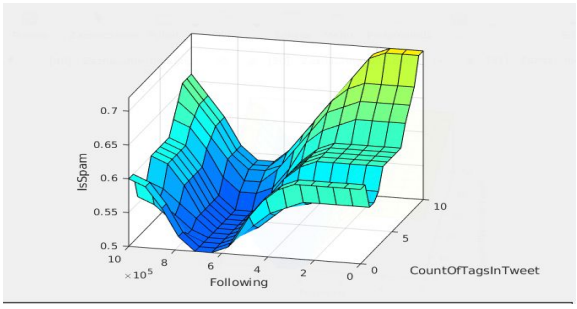
Z racji ilości reguł, będę prezentował tylko wyniki isSpam i NotSpam oraz wykresy surface (wymiary siatki 15x15 i 101 punktów)

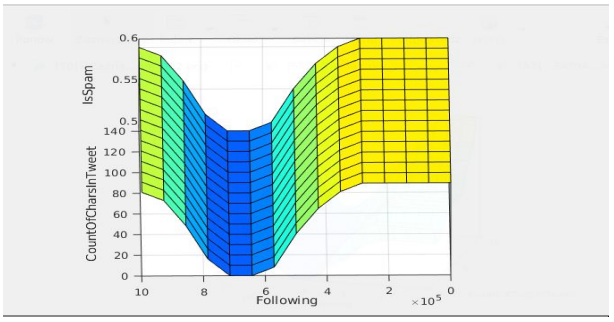


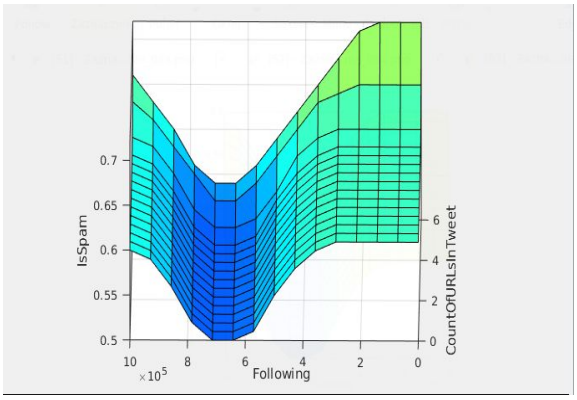
.

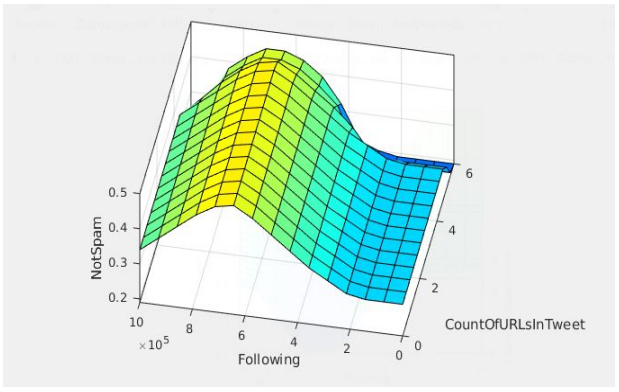

IsSpam
Following
Tweets

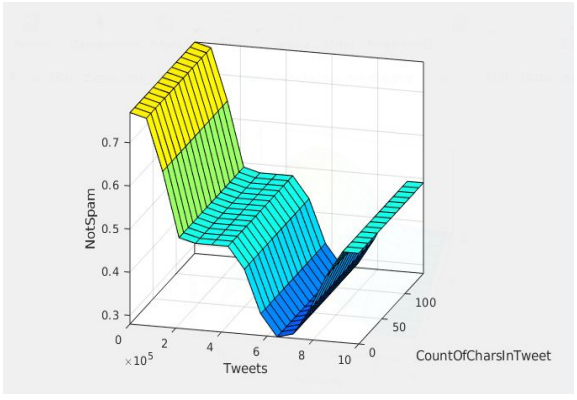

IsSpam
Following
CountOfWordsInTweet


IsSpam
Following
CountOfTagsInTweet

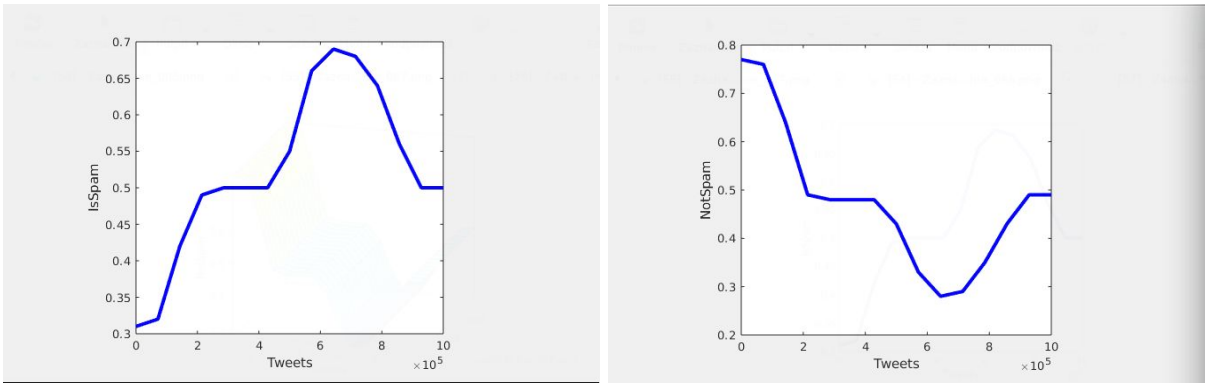

IsSpam
CountOfCharsInTweet
Following


IsSpam
Following
CountOfURLsInTweet

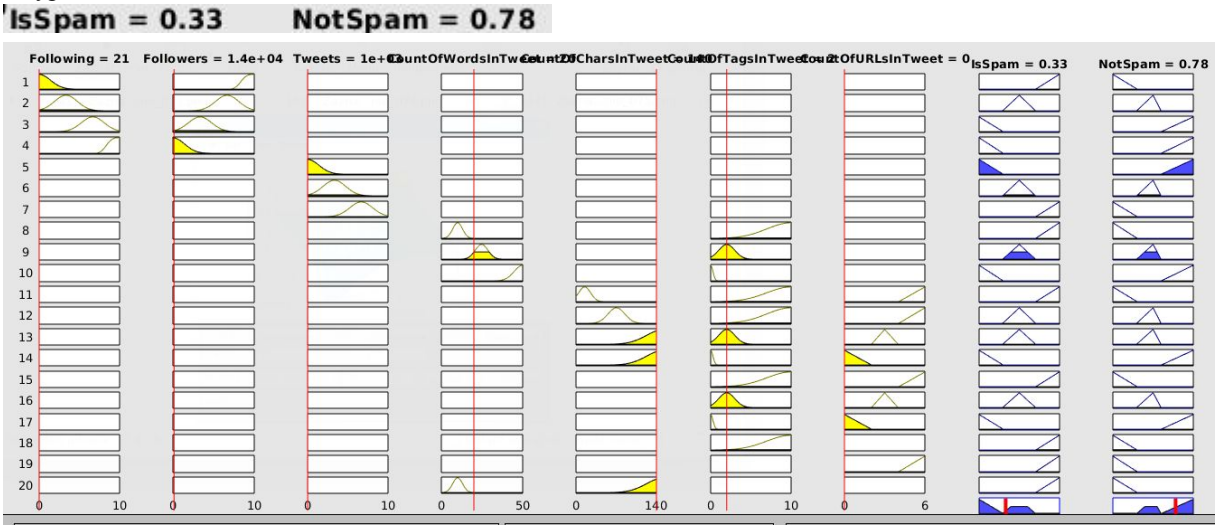

NotSpam
Following
CountOfURLsInTweet


NotSpam
Tweets
CountOfCharsInTweet

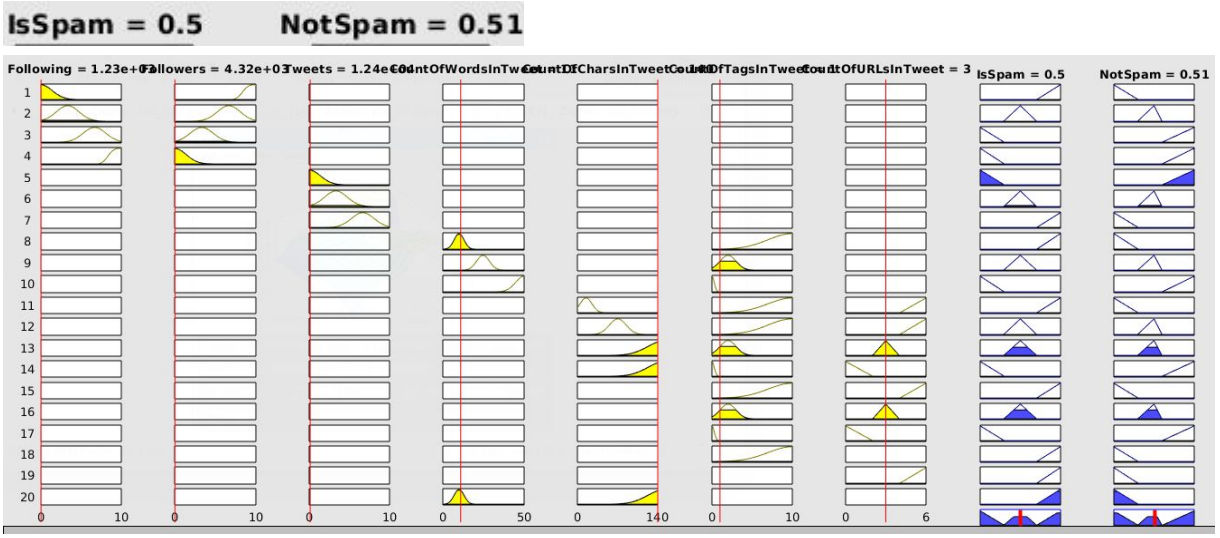
.



Przypadek 1:

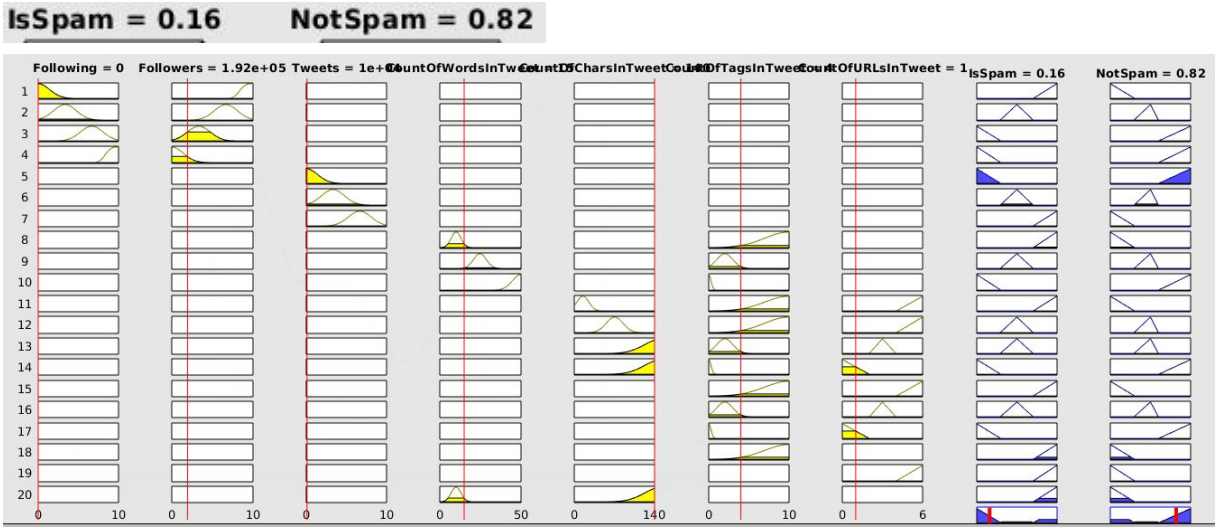


Przypadek 2:

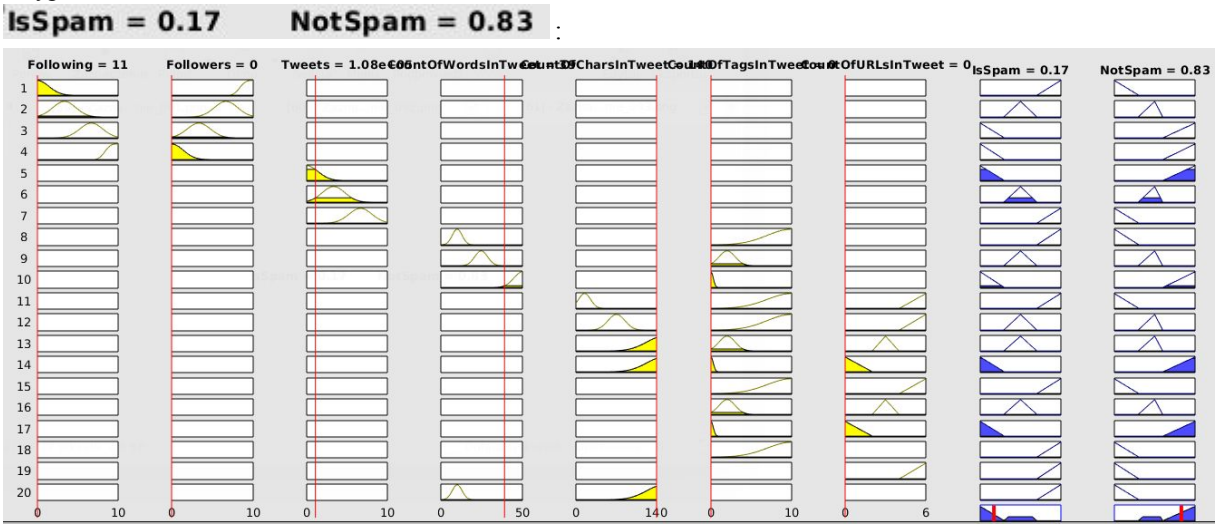


.

Przypadek 3:



Przypadek 4:



:

7. Podsumowanie i wnioski

Projekt w pełni realizuje zadania podstawowego klasyfikatora spamu, oraz posiada szerokie możliwości rozbudowania o dodatkowe funkcjonalności żeby w przyszłości nie służyć jako tylko i wyłącznie klasyfikator, ale detektor osób uciążliwych, ekspertów w danej dziedzinie, trolli internetowych czy też pomóc w tropieniu oszustw na portalach społecznościowych. Dodając więcej zmiennych wejściowych takich jak np. data utworzenia konta, jakość używanego słownictwa czy listę najczęściej używanych słów można by uzyskać wiele ciekawych wyników.

Wykorzystując idee obecnego projektu można w bardzo łatwy sposób rozszerzyć zastosowanie sterownika rozmytego o dowolny portal społecznościowy.

Obecnie, stworzony sterownik rozmyty mógłby znaleźć swoje zastosowanie w analizie szans kandydatów w wyborach (często przed wyborami pojawiają się konta, które generują dużo pochlebnych wiadomości wobec swojego kandydata oraz niepochlebnych wobec kontrkandydata) i chronić przed nachalną agitacją czy późniejsza propaganda (znany przypadek farm trolli w Ministerstwie Sprawiedliwości)

.

Zastosowanie w tym przypadku modelu Mamdaniego i stwierdzenie że nie ma jednomyślności czym jest spam, pozostawia otwarte drzwi do zmienienia koncepcji modelu i szukania zastosowania algorytmów **ANFIS** (Adaptive Neuro-Fuzzy Inference System) w celu wyindukowania reguł.