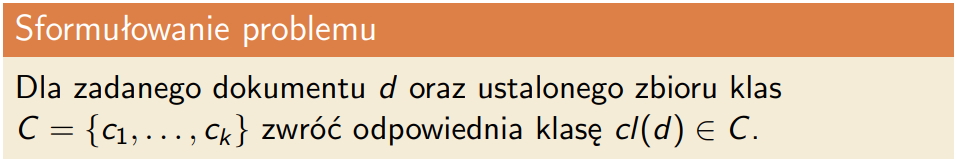
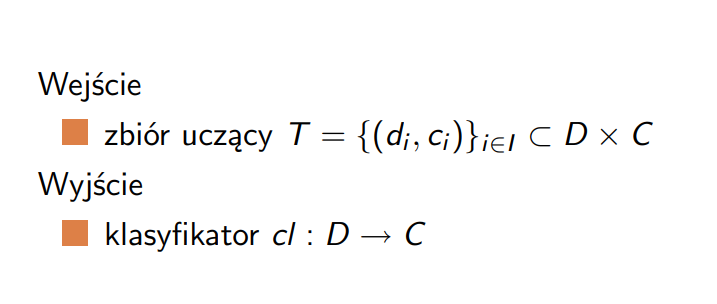
# Notatki NLP – wykład 20.10.2020

## **Klasyfikacje tekstu**



* Tu są różne sytuacje może być tylko 2 klasy dla każdeego dokumentu – taka binarna klasyfikacja (chyba tak jak na cw robiliśmy klasa negative i positive dla recenzji, może być wiele klas dla każdego dokuemntu.
* Naszym zadaniem jest więc znalezienie odpowiedniego modelu który będzie realizował funkcje, która dla każdego dokumentu będzie zwracała klasę lub najbardziej prawdopodobny podzbiór klas.
* Dobrym przykłądem z rzycia codziennego jest tak jak robilismy na cw ten filter spamu w mailach, lub własnie klasyfikacja dokumentów, identyfikacja płci
* Będziemy tutaj stosować podjeście statystyczne, będą własnie rozne rodzaje statystyk liczone
* Wejście – zbiór uczący – model
* Wyjście - klasyfikator
* Najprosztszą naiwną metodą reprezentacji naszego tekstu było by gdy weźmiemy sobie jakiś wektor np. 1000 elementowy i każdy element odpowiada jakiemuś słowu, i teraz jeżeli dany element (słowo) występuje w analizowanym tekscie to dajemy wartość 1 a jeżeli nie występuje to wartość 0 – TO SIĘ NAZYWA ONE-HOT ENCODING 🡪 sposób ten może być bardzo ograniczony bo no 1 lub 0 niesie bardzo niewiele informacji, ale wykrozystuje się to mimo wszystko
* Bag of words – potrzebujemy worek w którym znajdują się słowa które są dla naszego zadania naistotniejsze, a zatem – jeżeli naszym zadaniem jest rozpoznanie płci to na pewno w worku będzie szukanie rodzajów męskich i żeskich a nie będzie słów w formie nijakiej

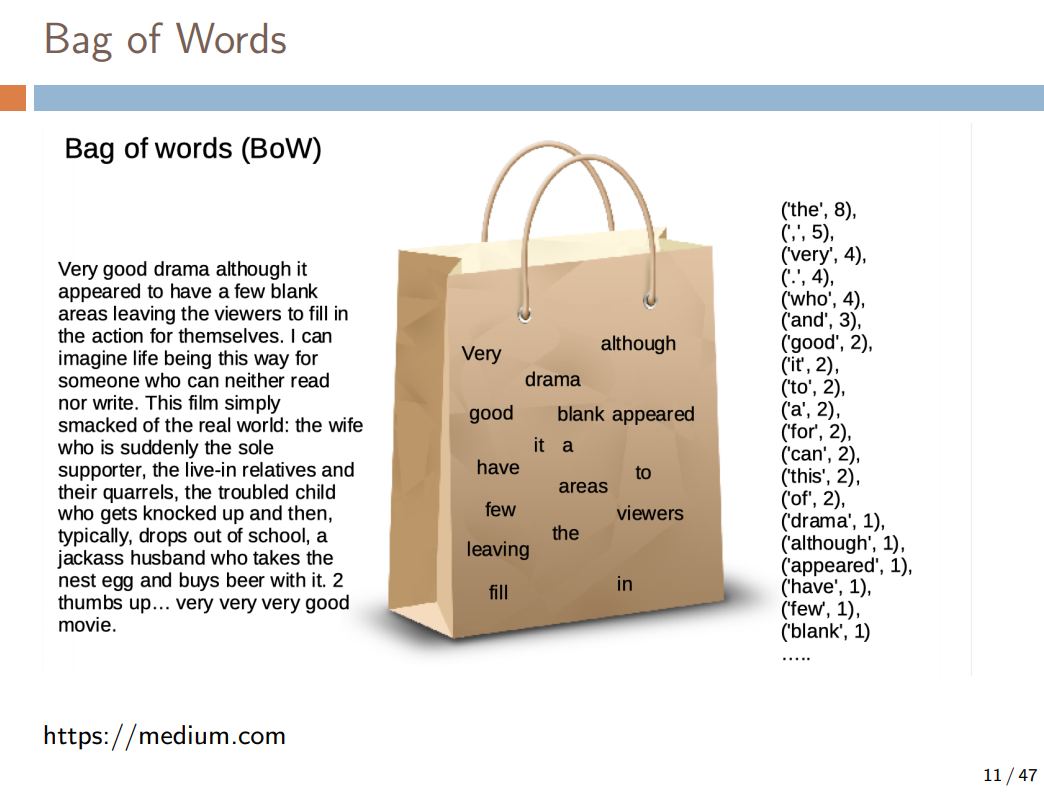
Tutaj też adrian troche inaczej tłumaczył i mówił że w worku są wszystkie słowa z wsyzstkich dokumentów i są potem zliczane po przepuszczeniu przez dokument

Wikipedia podaje :

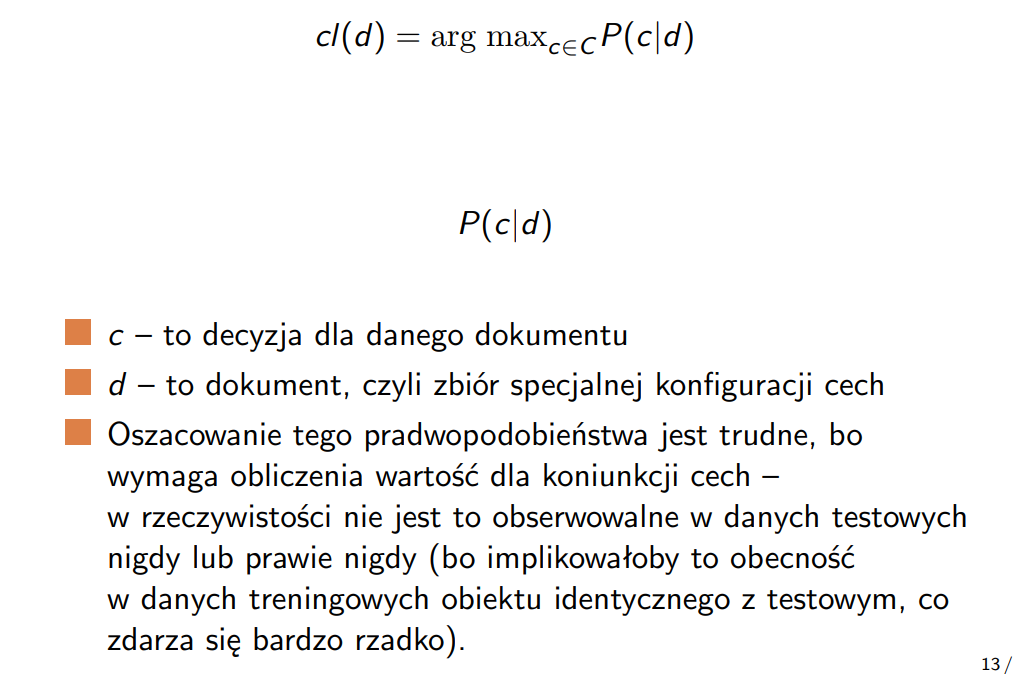
Model worka słów jest powszechnie stosowany w metodach klasyfikacji dokumentów, gdzie (częstotliwość) występowania każdego słowa jest wykorzystywana jako cecha do szkolenia klasyfikatora.

* 1. Czyli co robimy?

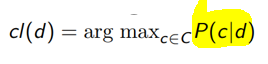
- Bierzemy nasze kolejne dokumenty i przepuszczamy kolejno przez worek słów, zliczamy wystąpienia każdego słowa w worku w dokumentach



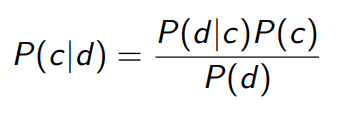
* Naive Bayes



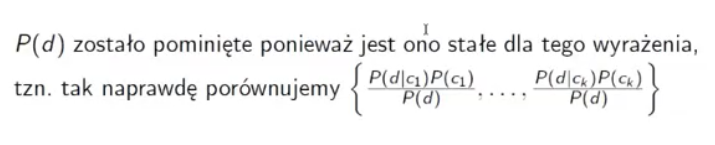
* Wracamy do naszego problemu i zastanawiamy się jak go można rozwiazać?
* Jednym z prostszych sposobów jest wyrazenie tego co wyzej czyli chcemy znaleść argumenty tej funkcji dla której osiąga ona swoje maksimum po wszystkich klasach w prawdopobiebienstwa przyporzadkowania do danej klasy danego dokumentu. Szacujemy prawdopobienstwo danej klasy pod warunkiem zaobserwownaia danego dokumentu.
* I tutaj z pomocą przychodzi nam BAYES bo to co robiliśmy na cw – twierdzenie bayesa nam fajnie rozwija tą część wzoru



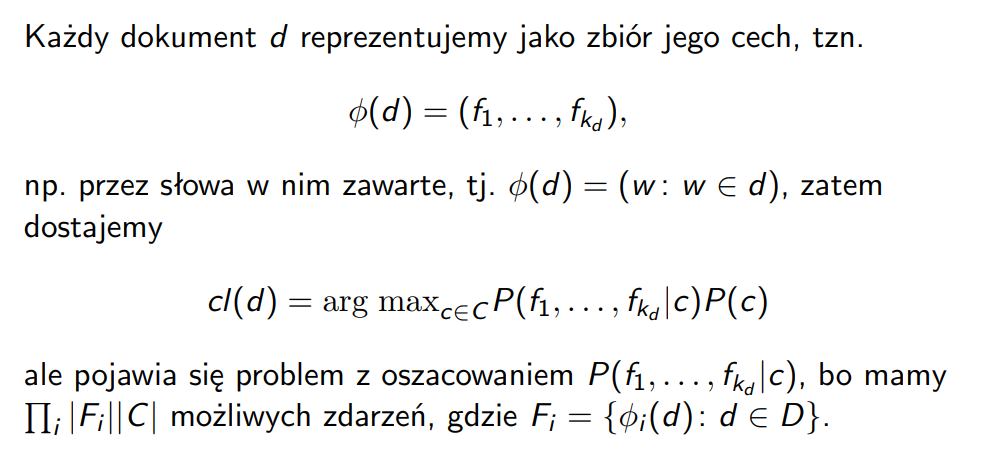
Tweirdzenie bayesa:

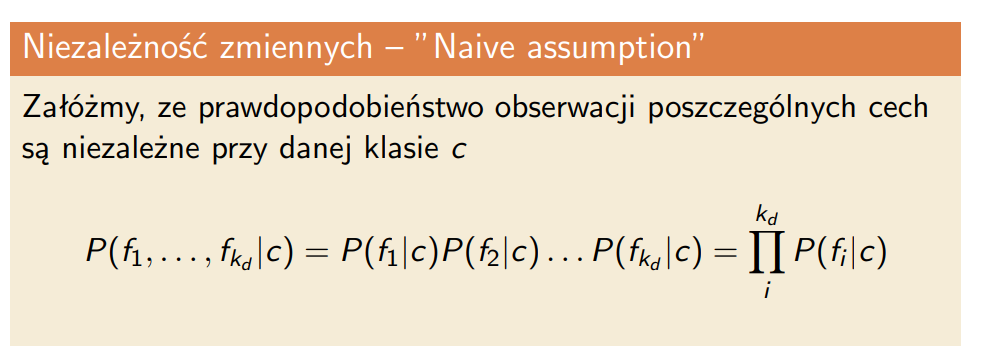


* Wymieniamy wiec prawdopodobienstwo klasy pod warunkiem dokumentu na iloraz gdzie w liczniku mamy prawdopodobientwo dokumentu w klasie razy prawdopodobienstwo klasy przez prawdopodobienstwo dokumentu (prawdopodobienstwo że zaobserwujemy dany dokument)

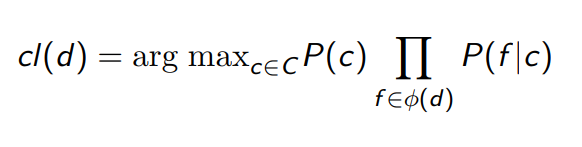


Popatrzmy na nasz wzór w którym szukamy maksimum po klasach a więc porównujemy jakby kolejne podstawienia do tej funkcji dla kolejncyh klas a wiec w mianowniku zawsze P(d) będzie takie samo wiec w sumie można je pominąć bo to co się będzie róznic tylko to licznik

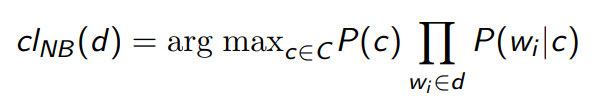
* Musimy już teraz wiedzieć tylko jakie jest prawdopodobienstwo klasy i prawdopodobienstwop dokuemntu pod warunkiem klasy
* Nasz dokuemnt możemy potraktwać najprościej jako występowanie pewnych słów (ogólnie cech), słowa te będą repezntacją naszego dokumentu
* Problem wtedy troche ze słowa koegzystują ze sobą ale w naszym naiwnym moddelu zakładamy że te słowa są niezależne i jak są niezależne to możemy wykorzystać Naive Bayes



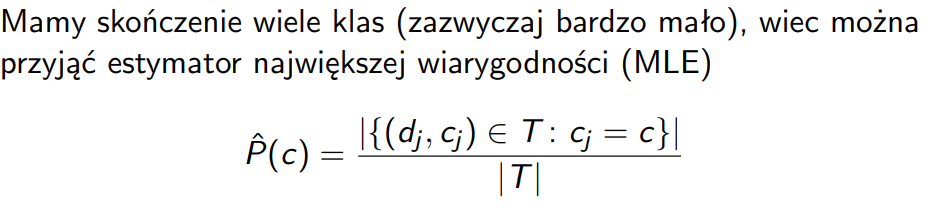
* I przy tych naiwnych założeniach nasz ostateczny wzór będzie wyglądał tak:

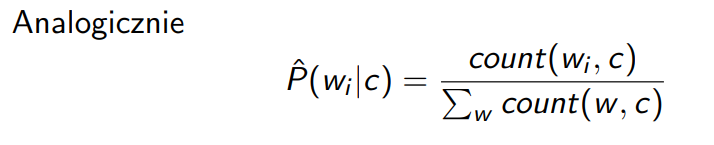


* Naszymi cechami mogą być wystąpeinia slów z bag of words



* Więc teraz nasze słowa wi będą pochodzić z bag of words
* Więc teraz szukamy prawdopodobienstwa slowa w klasie P(wi|c) i prawdopodobienstwo klasy P(c),
* Otóż P(c) liczymy tak: liczymy występowanie naszej klasy w naszym korpusie i podzielimy przez liczbę wszystkich dokumentów w naszym korpusie , co ma swoje ograniczenia bo nie wiemy jakie tam te teksty były jak były glównie pozytywne no to będzie to zaburzone



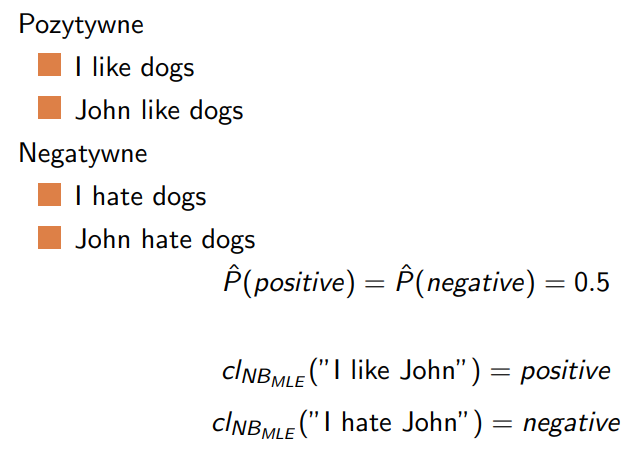
* Analogicznie liczymy prawdopodobienstwa wystąpienie słowa w klasie, liczymy ile razy słowo występuje w danej klasie przez (tutaj kurcze przerwy trochę ale no jest wzór XD) 
* Dwa linki do Bayesa:

<https://www.dailymotion.com/video/x2x5u4o>

<https://www.youtube.com/watch?gl=SN&hl=fr&v=1KOUpkU2cME>

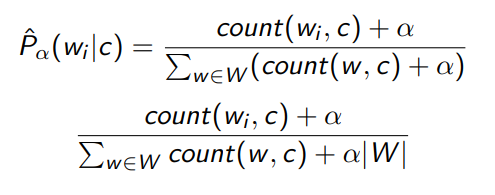
* I teraz przykłady, które Misztal chce żebyśmy umieli rozwiązywać na kartce!!!!

- Mamy dwie klasy pozytywną i negatywną – nasze korpusy nie są zbyt obszerne bo w każdej klasie mamy po dwa dokumenty:



* Prawdopodobienstwo klasy pozytywnej i negatywnej jest równe ½ (tutaj to zostało podane w tym przykładzie) ale tak jak w przykładzie adriana z ćw 20.11.2020 to może być np. 60 dokumentów pozytywnych i 40 negatywnych wtedy prawdopodobiesntwo klasy pozytywnej to 6/10.
* Możemy się natknąć na takie problemy – dostaniemy do zidentyfikowania zdanie które zawiera słowo które jest nowe i nie występowało w żadnej klasie np.: I like cats

Można zrobić takie „wygładzanie” (Laplace smoothing) wtedy w tych wzorach na prawdopodobienstwo dodamy w liczniku jakis wspólczynnik alfa i wtedy będziemy unikać zera (wohoo!) Bo z zerem to nic nie można zrobić niestety



* Dobra więc policzmy to! Czy przy takich klasach z korpusami jak poniżej zdanie „I like dogs” będzie pozytywne czy negatywne?

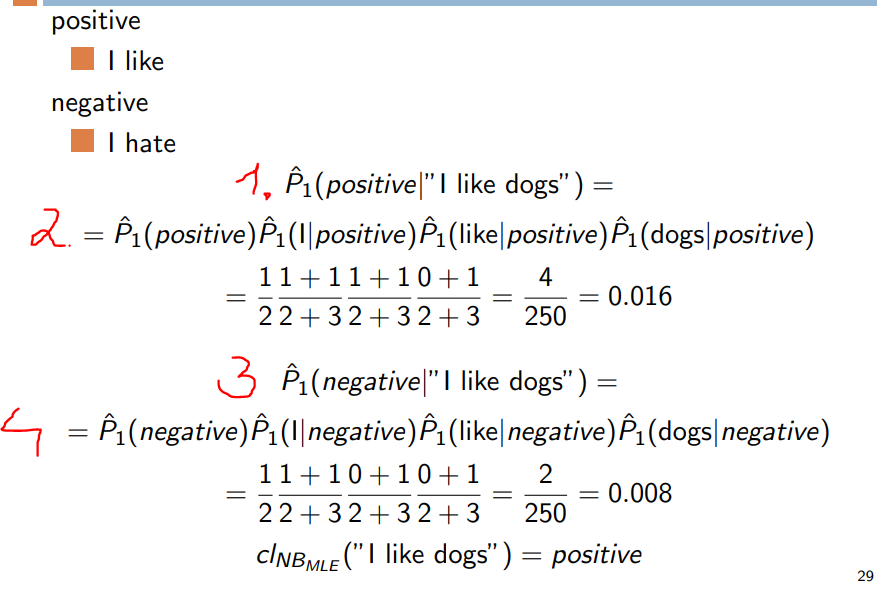
1. Najpierw trzeba policzyc prawdopodobienstwo klasy pozytywnej pod waruinkiem, że nasze zdanie to „I like dogs’
2. Żeby to policzyc to cyk to rozbijamy na wzorki:

- pradwopodobienstwo pozytywnej(1/2)\*prawdopodobienstwo wystąpienia słowa „I” pod warunkiem występowania klasy pozytwynej((1bo I występuje raz w słowniku+1bo to jest to wygładzenie żeby czasem w liczniku zera nie było)/(2bo 2 słowa w klasie+3bo długość słownika to 3”I, like, hate)) \*prawdopodobienstwo wystąpienia słowa „like” pod warunkiem występowania klasy pozytwynej((1bo like raz występuje w klasie+1bo wygładzenie)/(2bo 2 słowa w klasie+3bo długość słownika to 3”I, like, hate))\* prawdopodobienstwo wystąpienia słowa „dogs” pod warunkiem występowania klasy pozytwynej((0bo dogs nie występuje w słowniku+1to wygładzenie i dobrze ze jest bo tak to by w liczniku było 0 i byłby błąd)/(2bo 2 słowa w klasie+3bo długość słownika to 3”I, like, hate))

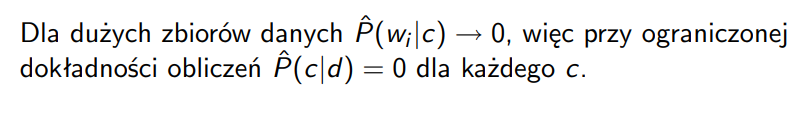
1. Teraz trzeba policzyc prawdopodobienstwo klasy negatywnej pod waruinkiem, że nasze zdanie to „I like dogs’
2. Żeby to policzyc to cyk to rozbijamy na wzorki:

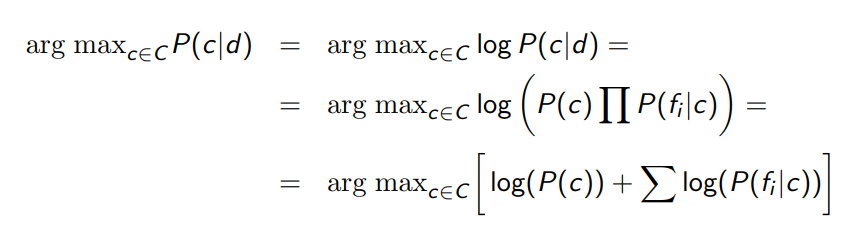
- pradwopodobienstwo negatywnej\*prawdopodobienstwo wystąpienia słowa „I” pod warunkiem występowania klasy negatywnej\*prawdopodobienstwo wystąpienia słowa „like” pod warunkiem występowania klasy negatywnej \* prawdopodobienstwo wystąpienia słowa „dogs” pod warunkiem występowania klasy negatywnej

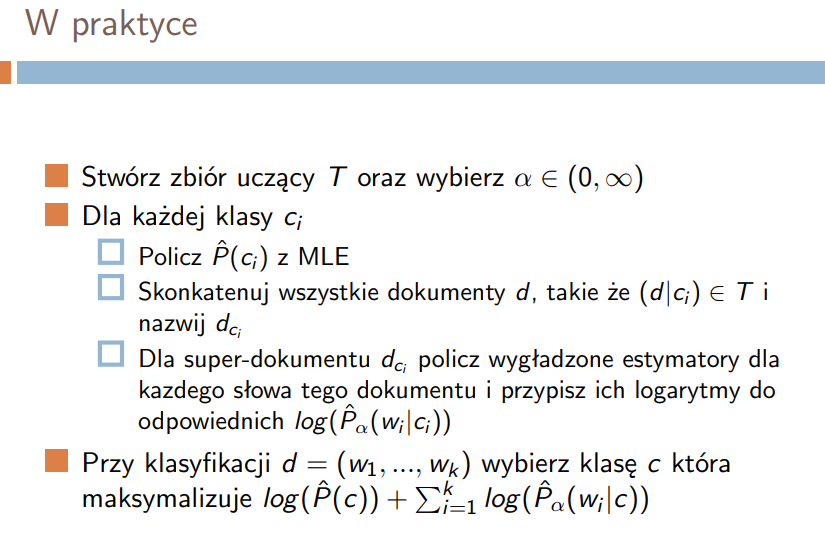
To samo co wyrzej w positive i wychodzi nam elegancko ze prawdopodobienstwo pozytywnej jest większe <3 wohoo



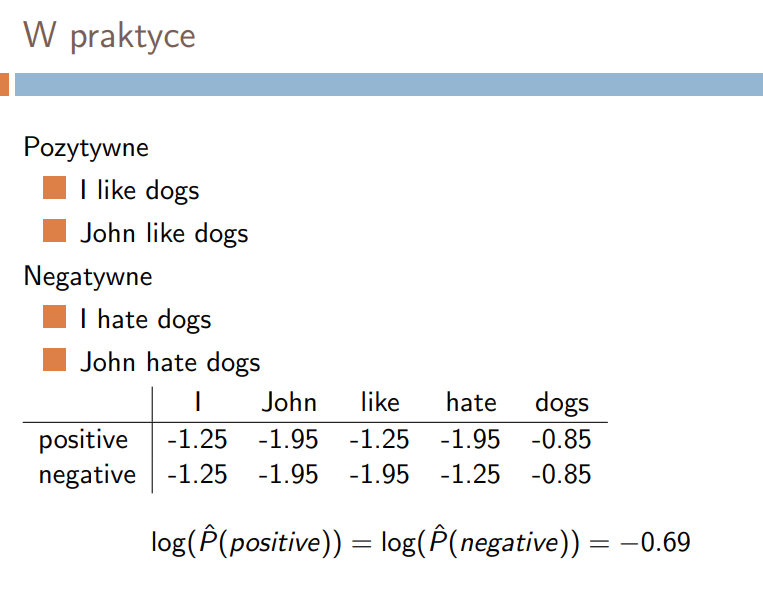
* Drugi problem bayesa – przy dużych danych, może się okazać że prawdopodobienstwo klasy będzie dążyło do zera więc, potem przy mnożeniu prawdopodobienstwa komputer nie będzie nam wstanie zreprezentować wyniku. Jak temu zaradzić? Cyk zamienic na logarytm i teraz te logarytmy można dodawać zamiast mnożyć i tak samo sprawdzamy gdzie wartosć jest największa, więc ponizej przykład

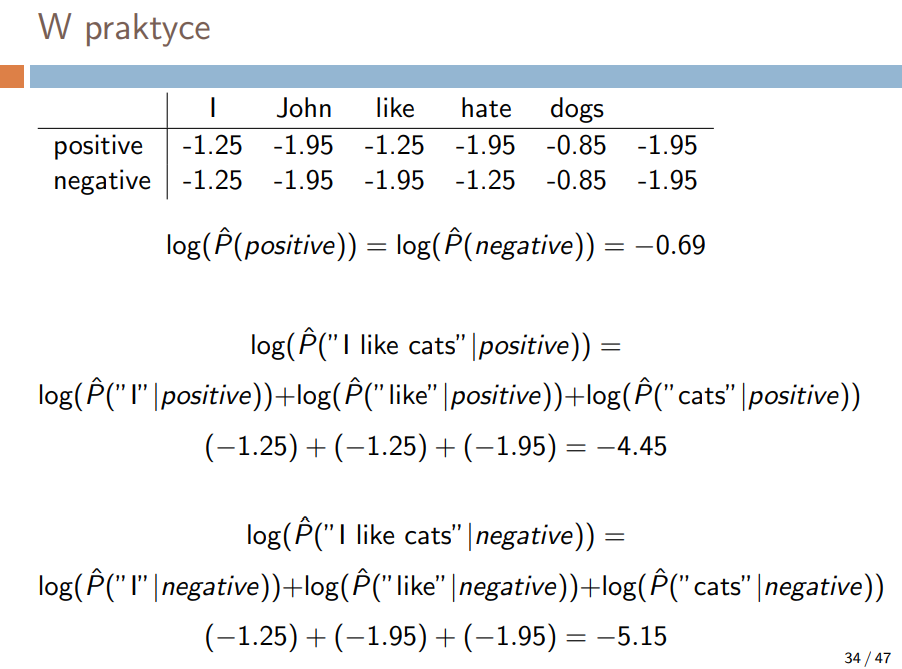


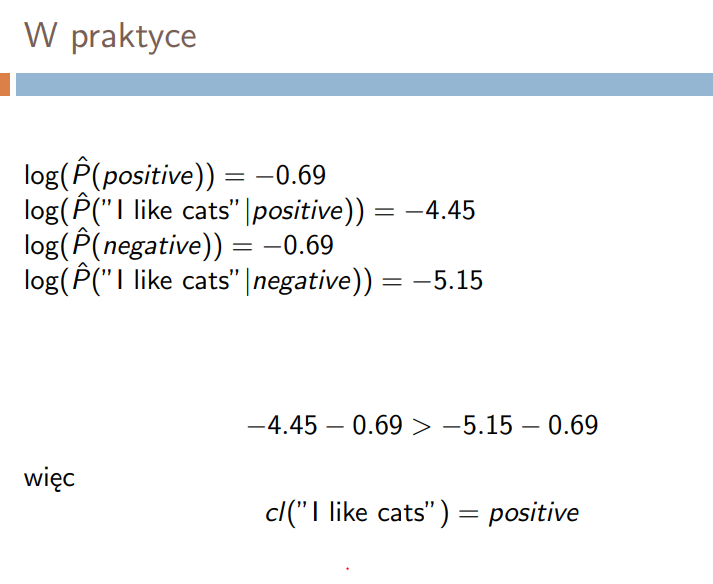




TO mamy sobie we własnym zakresie przesledzic XDD





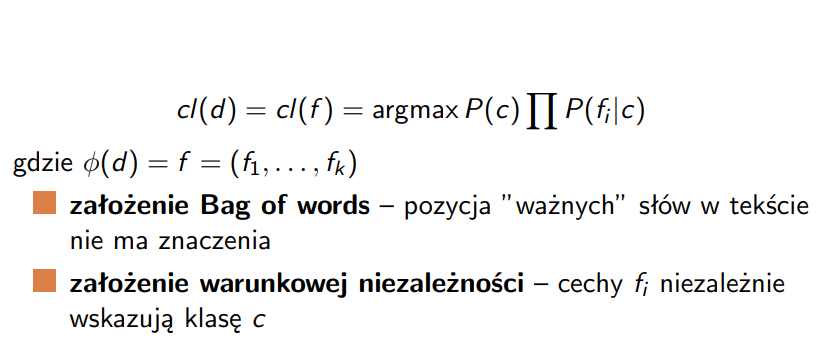


* Już teraz podsumowując –

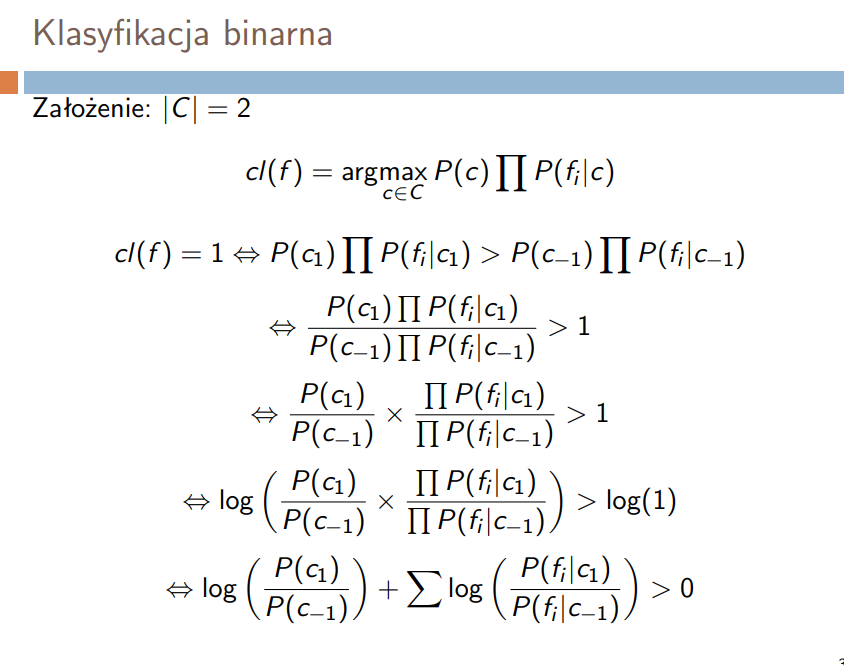
- należy pamietać mamy dwa założenia

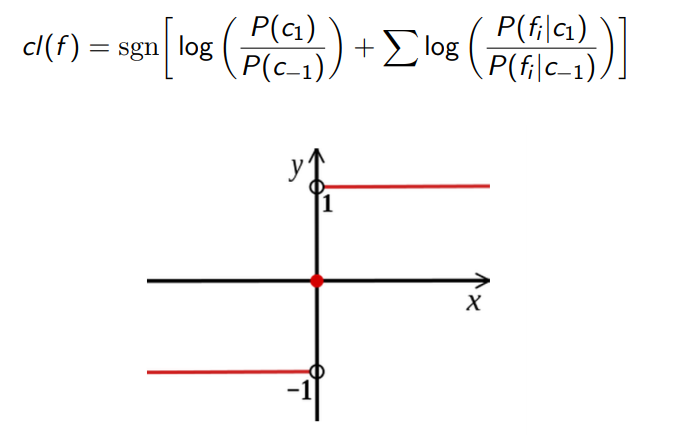
1. Reprezentujemy dokumenty jako bag of words

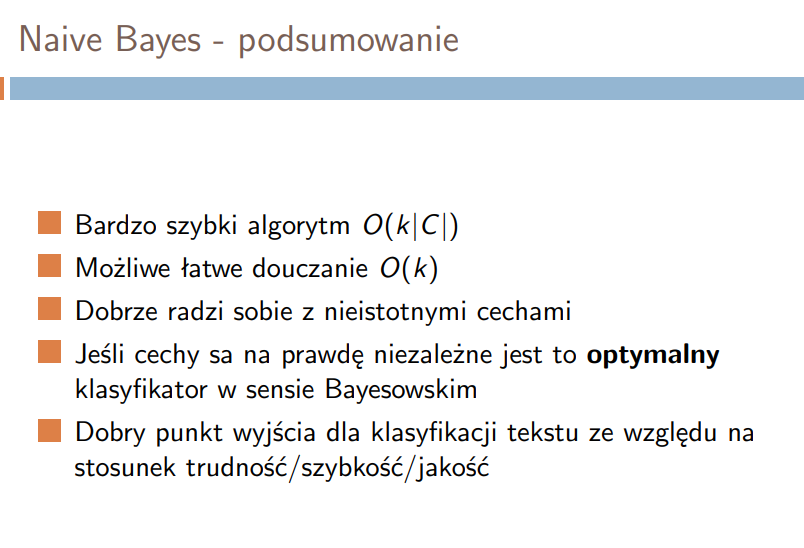
2. Nasze słowa są niezależne



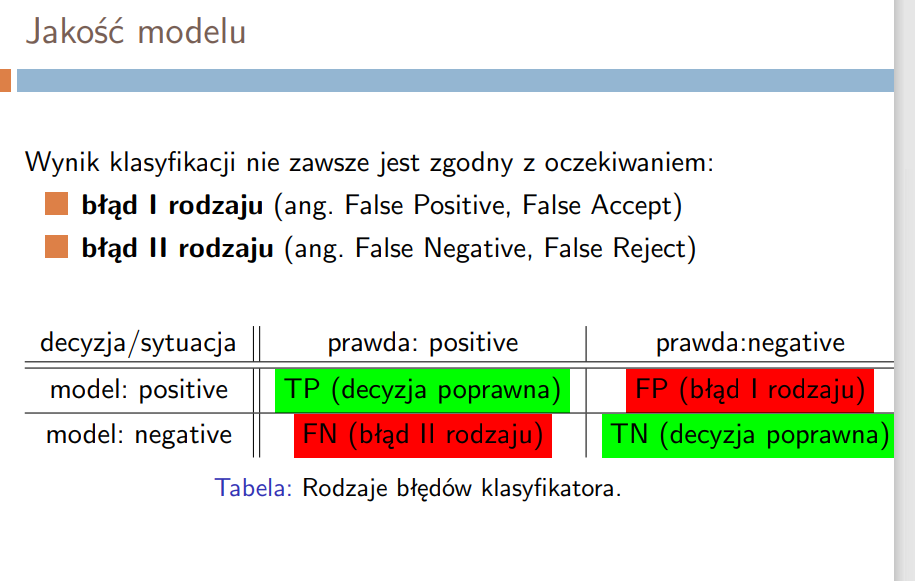
* W przypadku szczególnym gdy mamy tylko dwie klasy można to jakoś inaczej rozpisał ale już tutaj poleciał XD nie nie rozumeim , on mówi ze uproszczenie XD– jakby ktoś chciał posłuchać to 49 minuta ale to chyba nie jest aż takie ważne bo tak jak wczenisej też to można policzyć

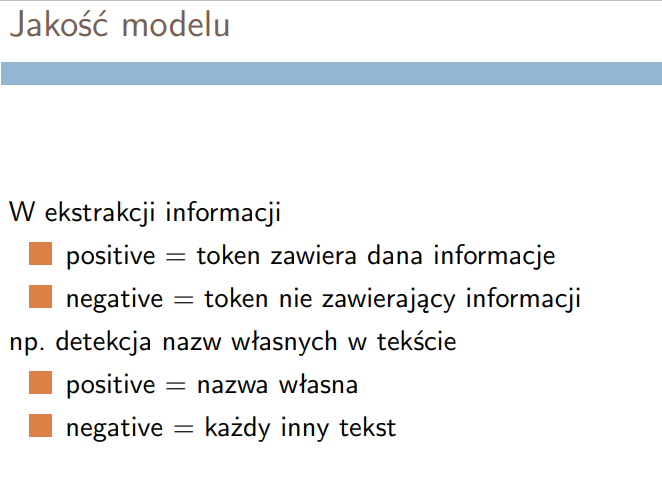


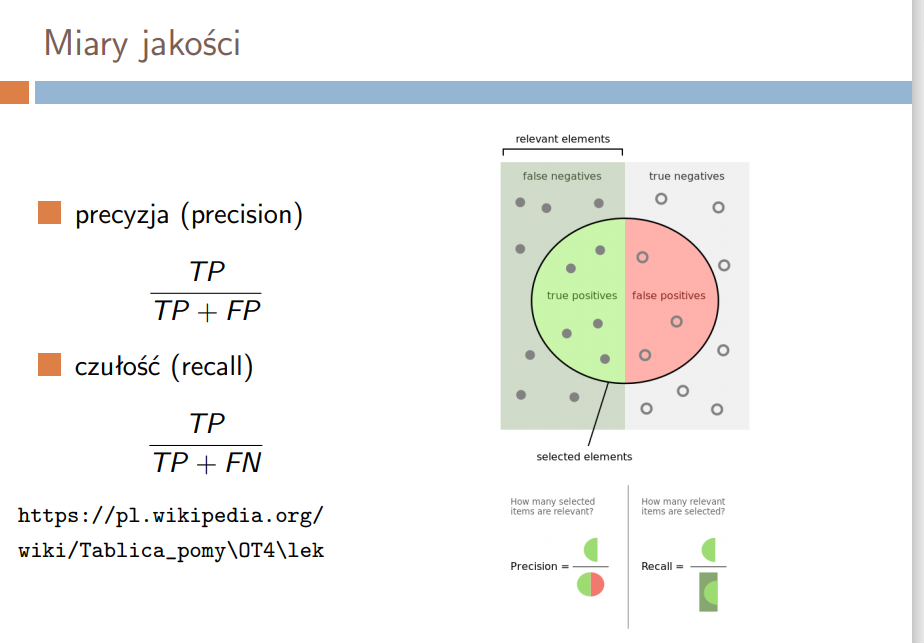




* Fajnie pozwalają na jakiś punkt odniesienia dla innych klasyfikatorów bo nie jest najlepszy ale jest szybki i jest to takie podstawowe narzędzie. Potem jakies klasyfikatory co sobie zrobimy to można z nim porówywać
* Jak ocenic jakość naszego modelu?
  + Podstawowym błędem jest uczenie i testowanie na tych samych danych BŁĄD!! bo testowanie na tym samym zbiorze co się uczyl to klasyfikator będzie super ale fajnie by było sprawdzic czy umie coś zrobić na nowym tekscie, albo dzielimy zbiór na połowe na jednym uczyc na drugim testować, albo lepiej podzielić zbiór na 5 czesci na 4 losowych uczyć a na 5 testowac wykonać to kilka razy i sprawdzic srednią







<https://sebastianraschka.com/Articles/2014_naive_bayes_1.html>

https://towardsdatascience.com/implementing-a-naive-bayes-classifier-for-text-categorization-in-five-steps-f9192cdd54c3