Politechnika Wrocławska
Wydział Informatyki i Zarządzania

Sprawozdanie

Sztuczna Inteligencja i Inżynieria Wiedzy
Uczenie maszynowe

Imię i nazwisko autora:	Adam Krzemiński
Nr indeksu:	246750
Data:	18.06.2021 r.
Semestr studiów:	6
Prowadzący laboratorium:	mgr. inż. Jan Jakubik

HR EXCELLENCE IN RESEARCH
Politechnika Wrocławska
Wybrzeże Wyspiańskiego 27
50-370 Wrocław
wiz.pwr.edu.pl

Spis treści

1. Opis analizowanego zbioru danych

Dane, jakie będą używane w zaimplementowanych systemach uczenia maszynowego pochodzą ze strony cs.cornell.edu. Zawierają one zestaw preprocesowanych recenzji filmów wraz z ich oceną punktową w skali 0-1. Recenzje, poza podziałem na autorów recenzji, który jest właściwie nieistotny, zawierają również przypisane do nich klasy na podstawie przyznanych ocen. Dla każdej z recenzji przypisana jest klasa w podziale ocen na 3 klasy oraz na 4 klasy. Tak pobrane dane będą ładowane do zaimplementowanego systemu, a następnie używane do klasyfikacji recenzji według przydzielonych klas.

2. Opis i cel zaimplementowanego systemu maszynowego uczenia się

W trakcie wykonywania tego laboratorium, do implementacji użyłem języka Python oraz wyspecjalizowanego w uczeniu maszynowym pakietu scikit-learn.

a. Cel laboratorium

Celem laboratorium było zapoznanie się i zbudowanie dwa systemy uczenia maszynowego do klasyfikacji tekstu, a konkretnie recenzji filmowych. Dwa podejścia, które należało zaimplementować to naiwny klasyfikator Bayesa oraz maszynę wektorów nośnych. Po zaimplementowaniu selekcji cech z recenzji, należało zbudować model obejmujący jeden z wyżej wymienionych klasyfikatorów, wytrenować go i wytestować. Ostatnim działaniem było przeprowadzenie różnych badań, których wyniki zawarte zostały w niniejszym sprawozdaniu.

b. Klasyfikator Naiwnego Bayesa

Jest to klasyfikator, który opiera swoje działanie i predykcję na obliczanych prawdopodobieństwach. Dodatkowo zakłada on, że wszystkie cechy są od siebie niezależne, stąd słowo „naiwny” w nazwie. Do klasyfikacji konkretnego, nowego tekstu wykorzystuje on dwa prawdopodobieństwa – a priori oraz a posteriori, obliczane globalnie i lokalnie. Mnożąc te dwa prawdopodobieństwa przez siebie, znajduje większe prawdopodobieństwo i do tej klasy przypisuje nowy tekst.

c. Maszyna Wektorów Nośnych SVM

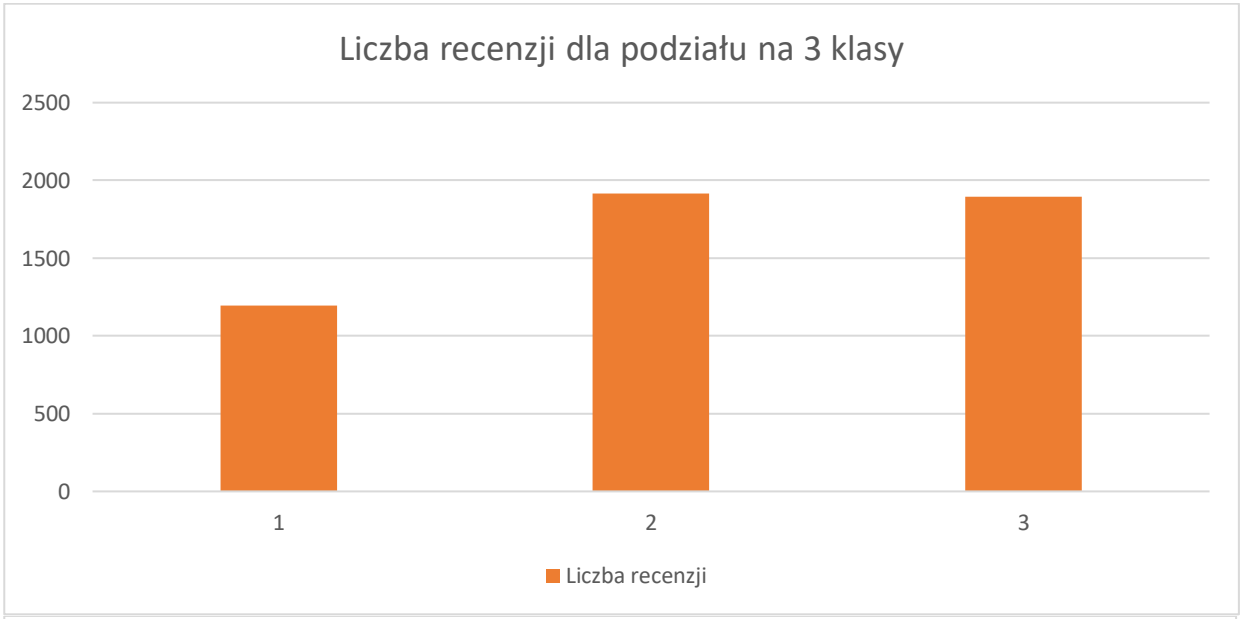
Maszyna wektorów nośnych jest klasyfikatorem, który polega na matematycznych algorytmach wyznaczenia płaszczyzny, która rozdziela obiekty przypisane do kilku klas. W ten sposób klasyfikuje on następnie nowe teksty.

3. Analiza eksploracyjna danych

W ramach prostych badań, przeanalizowałem kilka zależności, które mogą być przydatne np. w procesie implementacji selekcji cech.

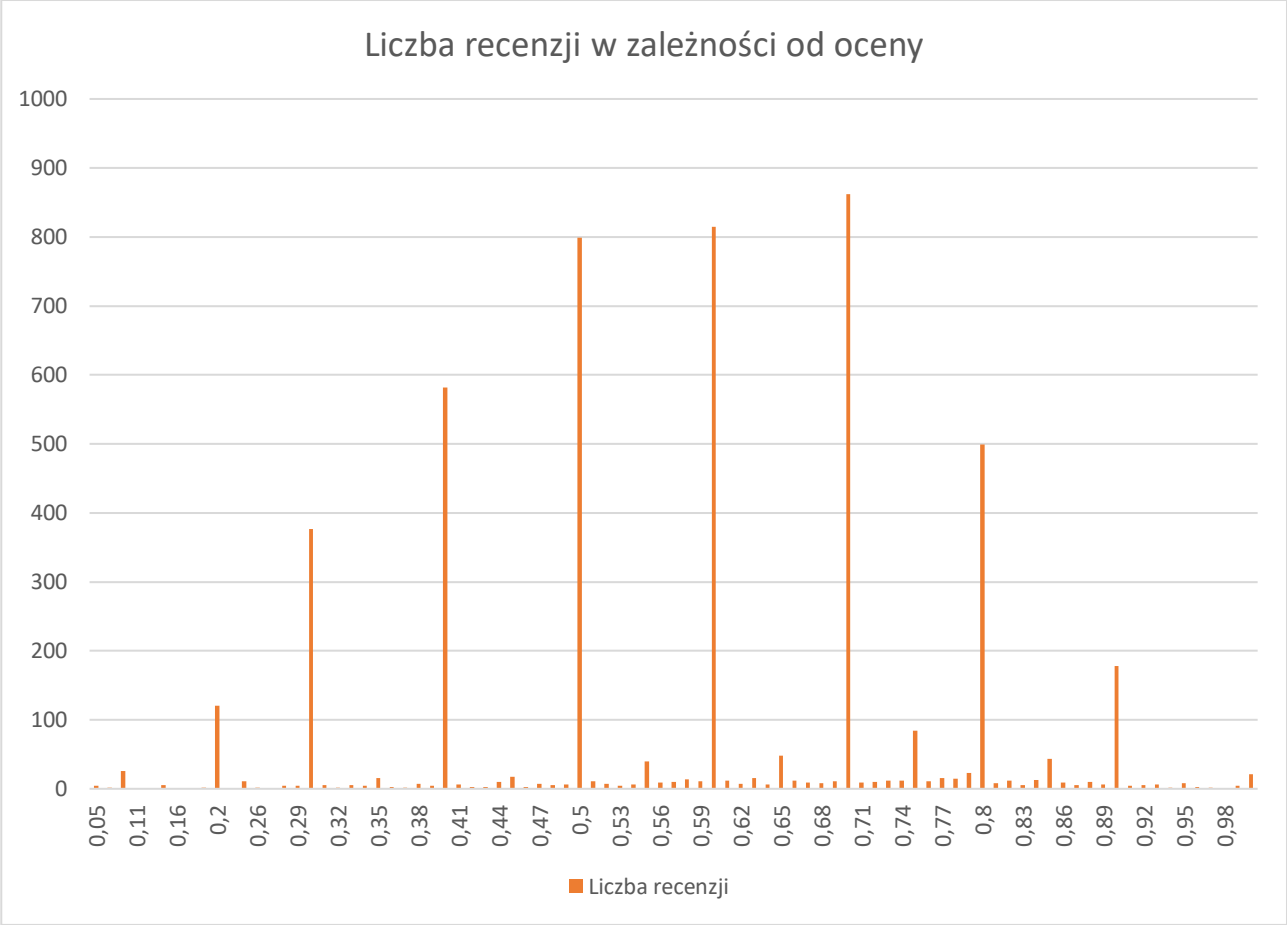
a. Liczba recenzji w zależności od klasy

Podział	Klasa	Liczba recenzji
3 Klasy	0	1197
	1	1915
	2	1894
4 Klasy	0	615
	1	1553
	2	1998
	3	840



Wnioski: Widać wyraźnie, zwłaszcza dla podziału na 4 klasy, że najwięcej jest recenzji o średniej ocenie, oceniający mniej chętnie przyznają oceny skrajne.

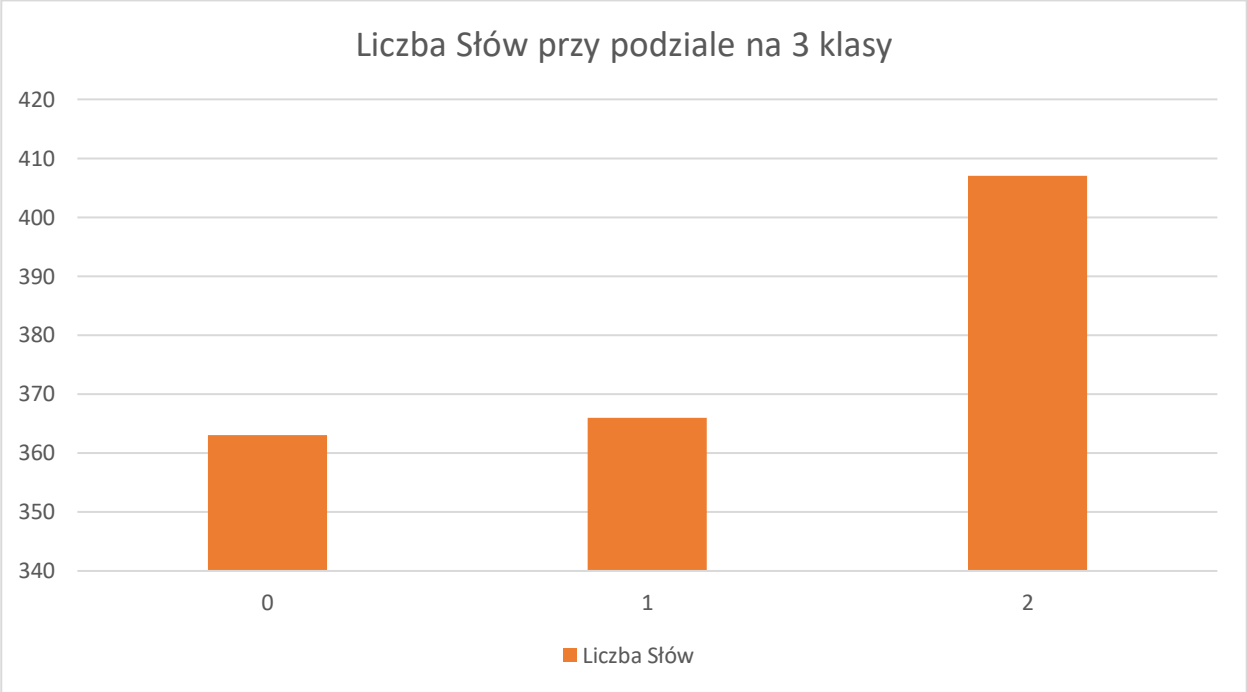
b. Liczba recenzji w zależności od oceny



Wnioski: Przy zastosowaniu wizualizacji liczby recenzji dla poszczególnych przyznanych ocen widać jeszcze bardziej, że dominują recenzje z ocenami środkowymi, rozkład przypomina kształtem rozkład normalny. Ponadto widać, że większość autorów recenzji ograniczała się raczej jedynie do ocen co 0,1; tylko jeden autor precyzował ocenę do jednej setnej.

c. Średnia liczba słów recenzji w zależności od klasy

Podział	Klasa	Średnia liczba słów
3 Klasy	0	363
	1	366
	2	407
4 Klasy	0	376
	1	356
	2	381
	3	428



Wnioski: W przypadku najlepszych recenzji są one zdecydowanie najdłuższe, widać, że w przypadku dobrego filmu autorzy recenzji chcą napisać o nim jak najwięcej.

d. Najczęściej występujące słowa w klasach

Podział na 3 klasy					
0 klasa		1 klasa		2 klasa	
word	**count**	**word**	**count**	**word**	**count**
film	4080	film	5951	film	5897
movie	2381	movie	3293	movie	3488
like	1609	like	2340	like	2248
story	1156	story	1998	story	1994
just	1075	time	1470	director	1549
time	880	just	1431	films	1533
director	848	director	1337	time	1493
good	788	good	1277	just	1493
films	773	little	1252	picture	1293
bad	732	characters	1194	characters	1276
make	711	films	1144	**best**	1214
characters	682	picture	1033	good	1105
little	639	make	944	little	1090
script	616	character	904	life	1046
look	517	way	900	character	1023
way	510	does	855	does	924
does	509	script	840	way	924
audience	482	**best**	753	script	812

Podział na 4 klasy							
0 klasa		1 klasa		2 klasa		3 klasa	
word	**count**	**word**	**count**	**word**	**count**	**word**	**count**
film	6173	film	11081	film	11773	film	22827
movie	3731	movie	5909	movie	7116	movie	1566
like	2465	like	4308	like	4644	like	980
story	1684	story	3686	story	3944	story	975
just	1653	just	2638	director	3093	director	765
time	1359	time	2595	films	3022	films	711
director	1301	director	2361	time	2884	time	708
good	1215	good	2236	just	2578	**best**	657
bad	1205	little	2229	picture	2574	just	653
films	1205	characters	2145	characters	2524	picture	608
make	1100	films	2044	good	2416	character	574
characters	1009	picture	1765	little	2342	life	554
little	947	make	1751	**best**	2177	good	471
script	945	character	1620	life	2054	characters	461
look	791	way	1576	character	1884	way	445
way	774	does	1556	does	1849	little	442
does	773	script	1537	way	1829	does	412
audience	736	better	1323	script	1654	great	386

Wnioski: Widać, że w przypadku najniższej klasy wysoko pojawia się słowo bad, a wraz z rosnącym numerem klasy zwiększa się pozycja słowa best w rankingu, które są związane z nacechowaniem danej recenzji. Pozostałe słowa są związane głównie z filmami i nie niosą za sobą jakiejś większej wartości.

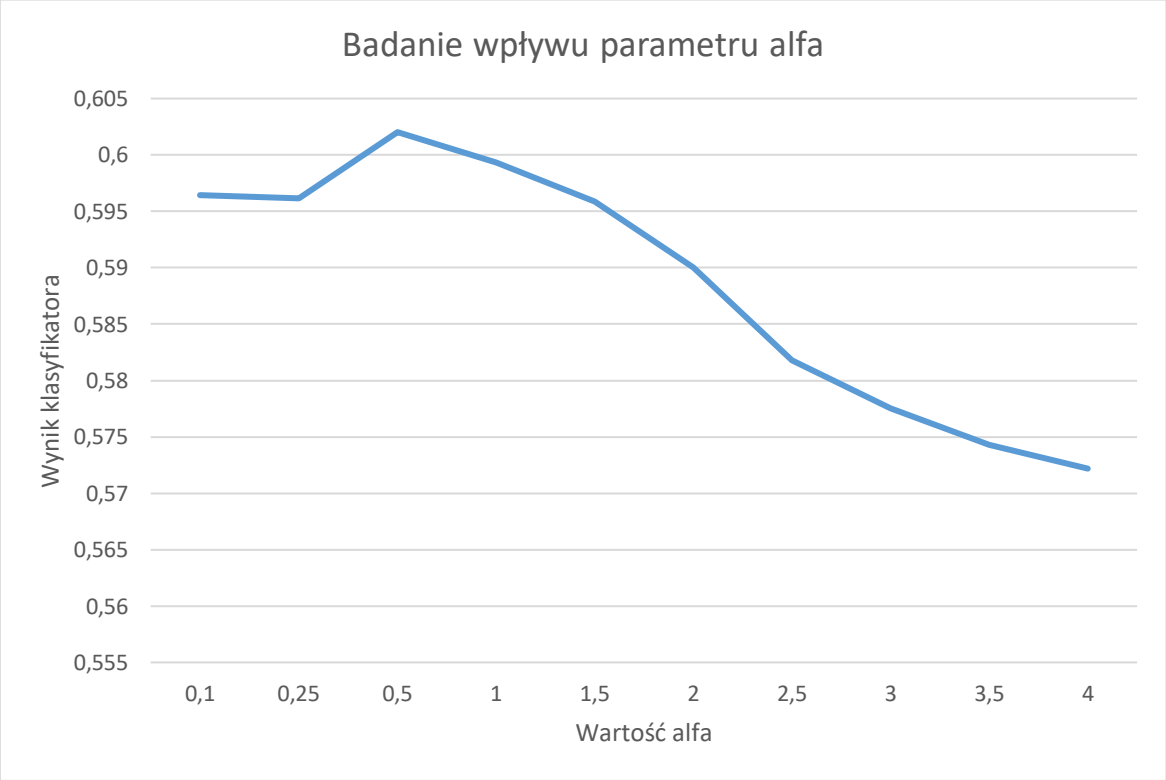
4. Przeprowadzone badania

Dla zaimplementowanych klasyfikatorów przeprowadziłem następujące badania:

a. Badanie wpływu hiperparametrów na klasyfikator Bayesa

Analizowałem różne wartości parametru alfa, zachowując stałą liczbę wyselekcjonowanych cech na poziomie 1000. Korzystałem z klasyfikatora MultiNomialNB. Dla badań dla podziału na 3 klasy otrzymałem następujące wyniki:

Alfa	Score
0,1	0,59641986
0,25	0,59615461
0,5	0,60202411
1	0,59935887
1,5	0,59589433
2	0,59003475
2,5	0,5817766
3	0,57751489
3,5	0,57431844
4	0,57218582

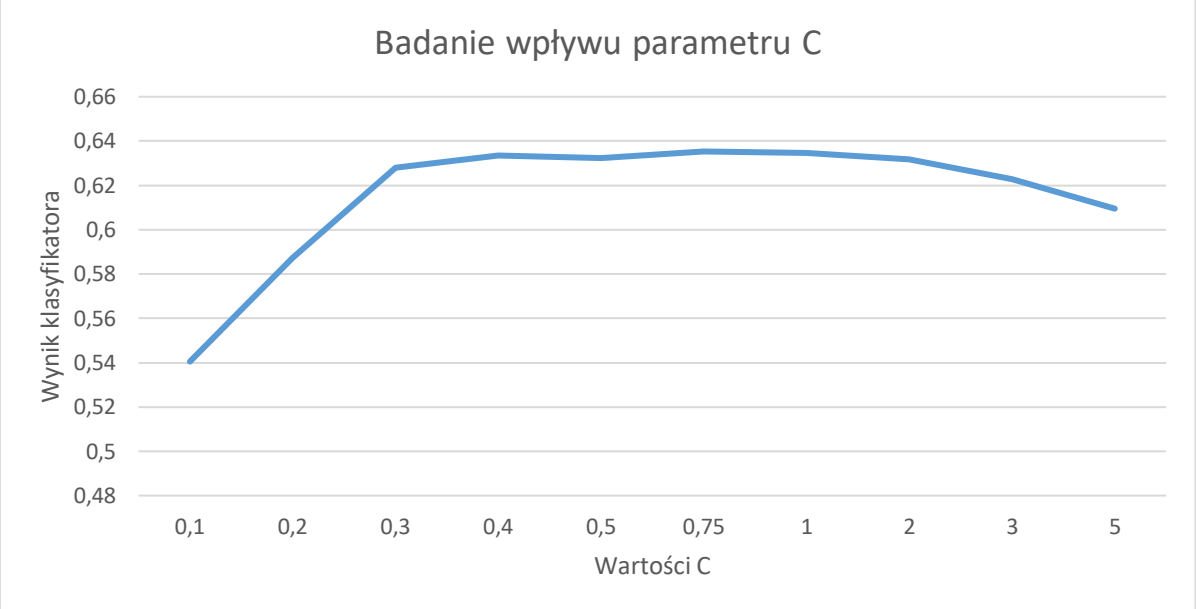


Wnioski: Parametr alfa ma wyraźny wpływ na wynik klasyfikatora Bayesa. Najlepsze wyniki osiąga dla wartości alfa około 0,5-1, a im większe jest alfa tym bardziej wyniki spadają.

b. Badanie wpływu hiperparametrów na maszynę SVM

Analizowałem różne wartości parametru C, zachowując stałą liczbę wyselekcjonowanych cech na poziomie 1000. Korzystałem z kernela linear. Dla badań dla podziału na 3 klasy otrzymałem następujące wyniki:

C	Score
0,1	0,540489
0,2	0,587359
0,3	0,628116
0,4	0,633442
0,5	0,632379
0,75	0,635305
1	0,634512
2	0,631848
3	0,62279
5	0,609472

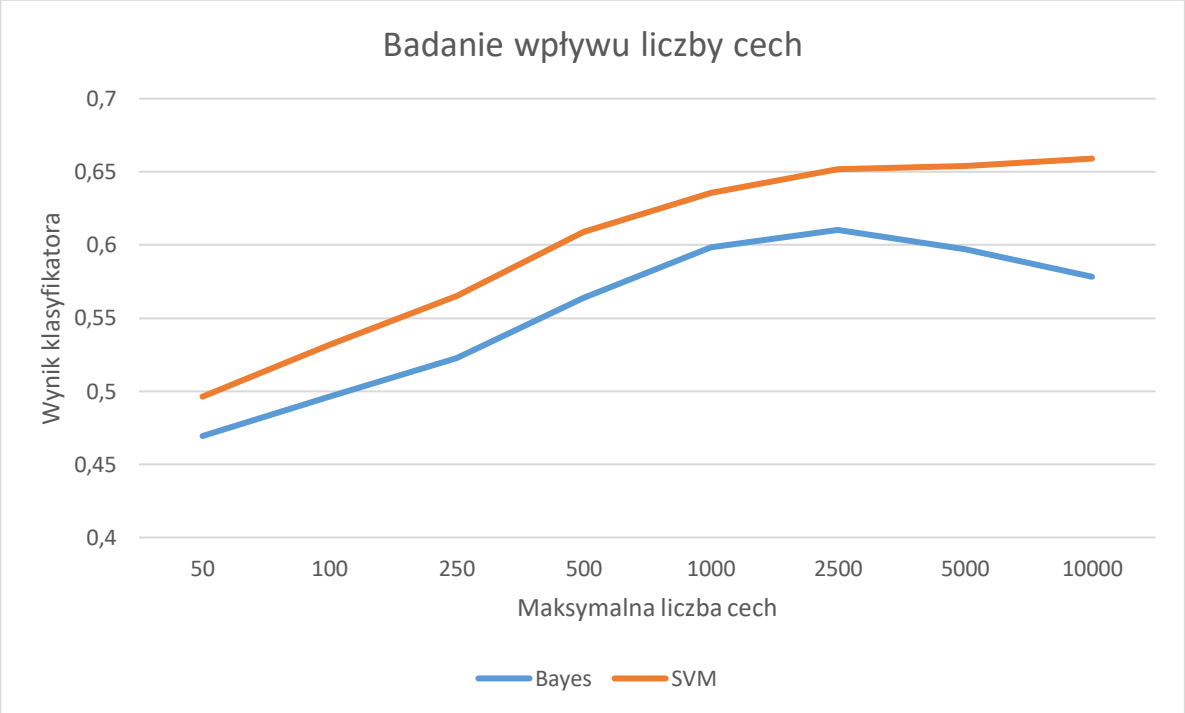


Wnioski: Parametr C ma wyraźny wpływ na wynik maszyny wektorów nośnych. Najlepsze wyniki osiąga dla wartości C między 0,3-2, dla bardzo małych wartości jej wynik jest bardzo mały, a gdy jest większe niż 2 to wraz ze wzrostem C również wyniki klasyfikacji spadają.

c. Badanie wpływu selekcjonowanej liczby cech na klasyfikatory

Analizowałem różne wartości parametru max_features dla selekcji cech, zachowując stałą wartość parametrów klasyfikatorów: alfa=1, C=1. Korzystałem z klasyfikatora MultiNomialNB, a dla maszyny SVM kernel ustawiony był jako linear. Dla badań dla podziału na 3 klasy otrzymałem następujące wyniki:

Max_features	Bayes Score	SVM Score
50	0,4693695	0,496282
100	0,49626667	0,531979
250	0,52263404	0,565025
500	0,56365603	0,608957
1000	0,59828865	0,635613
2500	0,61028156	0,651849
5000	0,5972156	0,653989
10000	0,57803546	0,659048

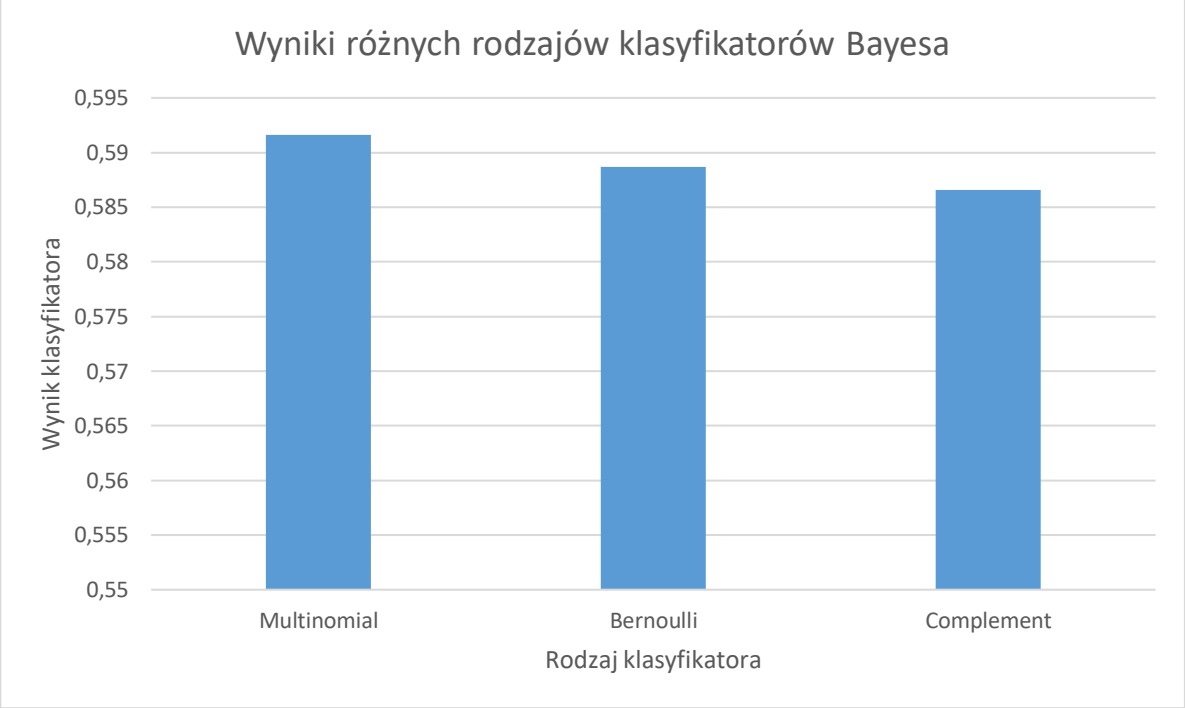


Wnioski: Bezpośrednie porównanie wyników pokazuje raz jeszcze, że wyniki klasyfikacji dla SVM są wyraźnie wyższe niż dla Bayesa. Co ciekawe, w przypadku SVM im wyższa liczba cech, tym wyższy wynik klasyfikacji, pomimo lekkiej stabilizacji wzrostu wyników od wartości 2500 cech. Z kolei dla klasyfikatora Bayesa najlepsze wyniki są dla 2500 cech, a powyżej tej wartości wyniki zaczynają spadać.

d. Badanie wpływu wybranych rodzajów klasyfikatorów Bayesa

Analizowałem różne rodzaje klasyfikatorów Bayesa, zachowując stałą wartość parametru alfa=1. Liczbę selekcjonowanych cech ustaliłem na poziomie 1000. Dla badań dla podziału na 3 klasy otrzymałem następujące wyniki:

Klasyfikator	Score
Multinomial	0,591614894
Bernoulli	0,588700709
Complement	0,586574468

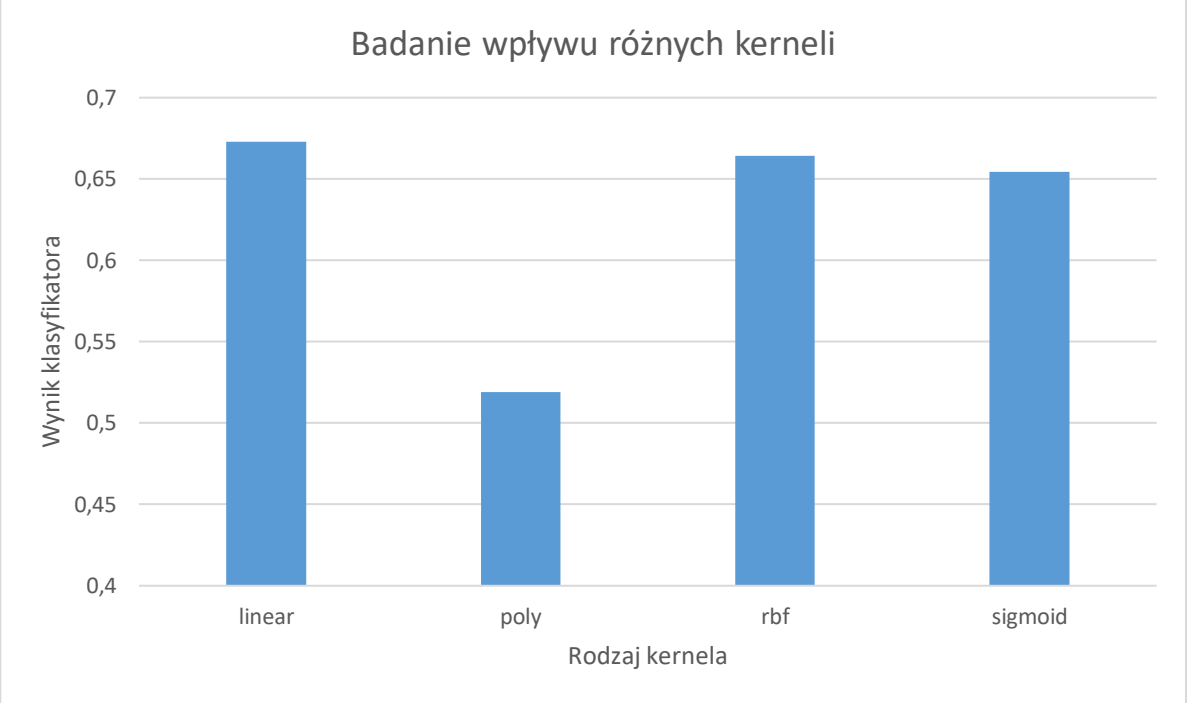


Wnioski: Wyniki klasyfikacji nie różnią się zbytnio pomiędzy różnymi rodzajami klasyfikatorów. Najwyższa jest jednak dla klasyfikatora Multinomial, co było do przewidzenia, gdyż jest on wyspecjalizowanym klasyfikatorem do klasyfikacji tekstu.

e. Badanie wpływu różnych kerneli dla SVM

Analizowałem różne rodzaje funkcji kernela dla maszyny SVM, zachowując stałą wartość parametru C=1. Liczbę selekcjonowanych cech ustaliłem na poziomie 1000. Dla badań dla podziału na 3 klasy otrzymałem następujące wyniki:

Kernel	Score
Linear	0,67288227
poly	0,519191489
rbf	0,664088652
sigmoid	0,654233333



Wnioski: Dla kernela poly wyniki klasyfikacji są znacznie i wyraźnie niższe, pozostałe kernele mają bardzo zbliżone do siebie wyniki. Spośród pozostałych, podobnych kerneli najwyższy wynik jednak osiągnął kernel linear, który, podobnie jak klasyfikator Multinomial Bayesa, jest dobrym kernelem do klasyfikacji tekstu.

5. Podsumowanie

Po wykonanych badaniach chciałem przedstawić wyniki najlepszych uzyskanych klasyfikatorów. Dlatego dla klasyfikatora Bayesa użyłem klasyfikatora MultiNomial i przeprowadziłem ostatnie testy dla najlepszych uzyskanych parametrów w badaniach:

Alfa\max_features	500	1000	2500	5000	10000
0,5	0,566582979	0,609202837	0,618789362	0,62999078	0,613730496
1	0,564452482	0,608937589	0,611335461	0,606813475	0,577507801
1,5	0,563121986	0,603346099	0,602279433	0,590031915	0,569785816

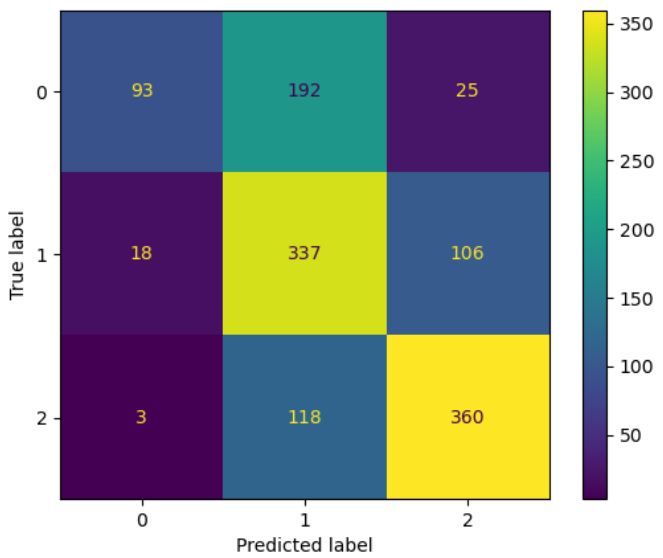
W ten sposób uzyskałem informację, że najlepszym z możliwych wariantów klasyfikatora Bayesa jakie przebadałem jest klasyfikator **MultiNomial**, z parametrem **alfa=0,5** oraz liczbą cech równą **5000**.

Natomiast dla maszyny wektorów nośnych (z kernelem ustawionym na funkcję linear) ostateczne testy zwróciły następujące wyniki:

C\max_features	500	1000	2500	5000	10000
0,5	0,61001773	0,654759574	0,660357447	0,66115461	0,652631206
0,75	0,607087234	0,653434043	0,664084397	0,672075177	0,668347518
1	0,604680851	0,644643972	0,660891489	0,669951773	0,672607092
2	0,602556738	0,637987943	0,639585816	0,651573759	0,656619149
3	0,60201844	0,634533333	0,63691773	0,647580851	0,652358865

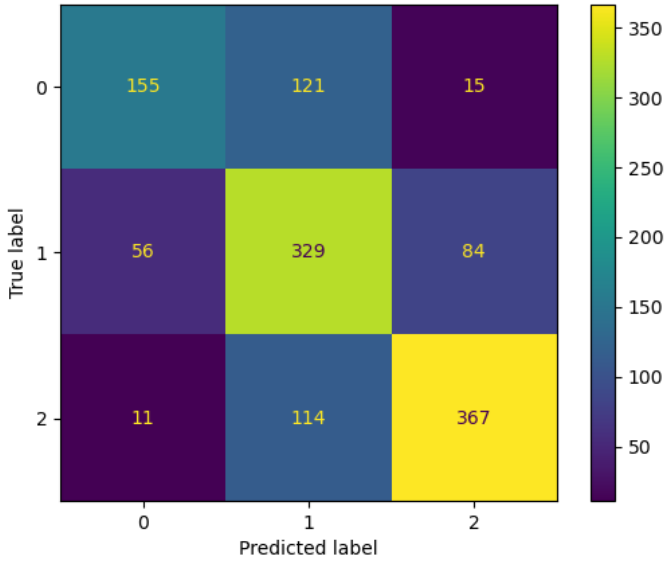
W ten sposób uzyskałem informację, że najlepszym z możliwych wariantów maszyny wektorów nośnych jakie przebadałem jest wersja z kernelem **linear**, z parametrem **C=1** oraz liczbą cech równą **10000**.
Przewaga tej wersji jest jednak minimalna nad maszyną o parametrach C=0,75 oraz liczbą cech równą 5000.

Po wytrenowaniu klasyfikatora Bayesa dla danych parametrów, dla danych testowych macierz klasyfikacji wygląda następująco:



Widać, że klasyfikator ten słabo radził sobie z klasyfikacją recenzji najsłabszych, zazwyczaj klasyfikował je do klasy środkowej. Również wyraźnie miał czasem problemy z rozróżnieniem klasy 1 i 2.

Natomiast po wytrenowaniu maszyny wektorów nośnych dla danych parametrów, dla danych testowych macierz klasyfikacji wygląda następująco:



Widać, że klasyfikator ten o wiele lepiej klasyfikował recenzje najsłabsze, lecz również miewał z tym problemy. Ponadto niektóre recenzje klasy 1 klasyfikował do najniższej klasy. Tak samo jak klasyfikator Bayesa miał czasem problemy z rozróżnieniem klasy 1 i 2.