Sztuczna inteligencja i inżynieria wiedzy

Sprawozdanie z zadania czwartego laboratorium

„Podstawy maszynowego uczenia na przykładzie klasyfikacji tekstu”

Prowadzący laboratorium:
dr inż. Paweł Myszkowski

Przygotował:
Aleksander Poławski

1. Wstęp

Celem zadania drugiego było zapoznanie się z metodami maszynowego uczenia poprzez zastosowanie ich w zagadnieniu klasyfikacji tekstu.

Skupiono się na dwóch metodach omawianych na wykładzie:
- naiwne klasyfikatory bayesowskie
- drzewa decyzyjne

Jako obiekty klasyfikacji wybrano dokumenty tekstowe, a jako cel klasyfikacji przypisanie ich do określonych z góry ustalonych kategorii tematycznych.

Jako wzorcową kolekcję etykietowanych danych przyjęto podzbiór artykułów z portalu „Wikipedia” z przypisanymi kategoriami z tego samego portalu. Zestaw składa się z liczby 9837 dokumentów zaklasyfikowanych do 34 różnych etykiet (kategorii tematycznych).

2. Selekcja i ekstrakcja cech

Dokumenty tekstowe jako obiekty klasyfikacji charakteryzują się bardzo dużą liczbą potencjalnych cech. Popularnym rozwiązaniem jest konwersja dokumentów do ciągów zawierających informacje o tym które słowa się w nich znajdują lub informacje o liczności poszczególnych zawartych w nich słów.

We wzorcowym zbiorze blisko 10 tysięcy dokumentów znajduje się jednak powyżej 300 tysięcy unikalnych słów i przyjęcie ich wszystkich jako cech uniemożliwiłoby poprawne działanie klasyfikatorów.

Opracowano program w języku C#, którego celem jest selekcja najbardziej przydatnych cech(słów) w klasyfikacji dokumentów do wzorcowych kategorii oraz ich ekstrakcja dla wzorcowego zbioru dokumentów.

a) Etap selekcji:

- program określa za pomocą algorytmu Mutual Information (MI) relewantność każdej z 300 tysięcy potencjalnych cech (słowa o minimalnej długości 4) dla każdej z możliwych klas

- następnie potencjalne cechy sortowane są ze względu na ich wskaźnik MI dla każdej klasy

- ostatecznie jako zbiór wyselekcjonowanych cech przyjmowana jest suma zbiorów L cech o najwyższym wskaźniku MI dla każdej z klas

- liczba wyselekcjonowanych cech jest równa liczbie L * ilość możliwych klas. Istnieje możliwość, że cecha zostanie uznana jako relewantna dla więcej niż jednej klasy, w tym wypadku liczba ostatecznie wyselekcjonowanych cech może być mniejsza (wariant mało prawdopodobny).

- L jest parametrem wejściowym programu (metody selekcji)

Wskaźnik MI dla danej klasy i cechy wyznaczono na podstawie wzoru:

$$I(U;C) = \frac{N_{11}}{N}\log_2\frac{NN_{11}}{N_{1.}N_{.1}} + \frac{N_{01}}{N}\log_2\frac{NN_{01}}{N_{0.}N_{.1}} + \frac{N_{10}}{N}\log_2\frac{NN_{10}}{N_{1.}N_{.0}} + \frac{N_{00}}{N}\log_2\frac{NN_{00}}{N_{0.}N_{.0}}$$

, gdzie:
N11 – liczba dokumentów klasy w których znajdowała się rozpatrywana cecha,
N10 – liczba dokumentów innych klas w których znajdowała się rozpatrywana cecha,
N01 – liczba dokumentów klasy w których dana cecha nie występowała
N00 – liczba dokumentów innych klas w których dana cecha nie wystąpiła
N0. – suma N01 i N00
N1. – suma N10 i N11
N.0 – suma N10 i N00
N.1 – suma N01 i N11
N – suma N11, N01, N10, N00

Poniżej załączono zrzut ekranu z przykładowego uruchomienia programu, ukazujący dziesięć wyznaczonych, potencjalnie najbardziej relewantnych cech oraz odpowiadające im wskaźniki MI dla kategorii „Amerykanscy-prozaicy”:

```
[Amerykanscy-prozaicy]
feature = [powieści]  MI = [0,104746223067685]
feature = [pisarz]  MI = [0,0766529251843881]
feature = [fiction]  MI = [0,0598883262239407]
feature = [amerykański]  MI = [0,0597331513992801]
feature = [opowiadań]  MI = [0,0584779212362473]
feature = [fantasy]  MI = [0,053715732786775]
feature = [science]  MI = [0,0515362315936022]
feature = [powieść]  MI = [0,0468585924019626]
feature = [opowiadania]  MI = [0,0425773314922562]
feature = [twórczość]  MI = [0,0390342704467293]
```

b) Etap ekstrakcji:

- dla każdego dokumentu zliczana jest ilość wystąpień każdej z wyselekcjonowanych cech

- zebrane informacje zapisywane są do pliku w formacie ARFF składającego się ze specyficznego dla tego formatu nagłówka, spisu wszystkich możliwych atrybutów (cech) i sekcji zawierającej tablice rzadkie informujące o liczności wystąpień poszczególnych cech w każdym z dokumentów

- zastosowano tablice rzadkie („sparse”) w celu zmniejszenia czasu zapisywania i ograniczenia wielkości pliku

Poniżej załączono zrzut ekranu przedstawiający fragment przykładowego, wygenerowanego pliku ARFF ukazujący jego nagłówek oraz początek sekcji definicji atrybutów (cech):

features_335.arff | features_335.arff | features_335.arff

```
% 1. Wyekstrahowane cechy dokumentów do ich dalszej klasyfikacji.
%2. Zajęcia: Sztuczna Inteligencja i inżynieria wiedzy
%3. Autor: Aleksander Poławski

@RELATION wikipedia_texts

@ATTRIBUTE class {Albania, Amerykanscy-prozaicy, Arabowie, Astronautyka, Choroby,
 Egipt, Ekologia-roslin, Filmy-animowane, Galezie-prawa, Gry-komputerowe,
 Karkonosze, Katolicyzm, Komiksy, Komputery, Kotowate, Kultura-Chin, Monety,
 Muzyka-powazna, Narciarstwo, Narkomania, Niemieccy-wojskowi,
 Optyka, Pierwiastki-chemiczne, Pilka-nozna, Propaganda-polityczna,
 Rachunkowosc, Samochody, Samoloty, Sporty-silowe, System-opieki-zdrowotnej-w-Polsce,
 Szachy, Wojska-pancerne, Zegluga, Zydzi}
@ATTRIBUTE feat_albanii NUMERIC
@ATTRIBUTE feat_albański NUMERIC
@ATTRIBUTE feat_tiranie NUMERIC
@ATTRIBUTE feat_albańskiej NUMERIC
@ATTRIBUTE feat_albańskiego NUMERIC
@ATTRIBUTE feat_obwodzie NUMERIC
@ATTRIBUTE feat_albańska NUMERIC
@ATTRIBUTE feat_komuna NUMERIC
@ATTRIBUTE feat_fabularny NUMERIC
@ATTRIBUTE feat_administracyjnie NUMERIC
@ATTRIBUTE feat_powieści NUMERIC
@ATTRIBUTE feat_pisarz NUMERIC
@ATTRIBUTE feat_fiction NUMERIC
@ATTRIBUTE feat_amerykański NUMERIC
@ATTRIBUTE feat_opowiadań NUMERIC
@ATTRIBUTE feat_fantasy NUMERIC
@ATTRIBUTE feat_science NUMERIC
@ATTRIBUTE feat_powieść NUMERIC
@ATTRIBUTE feat_opowiadania NUMERIC
@ATTRIBUTE feat_twórczość NUMERIC
@ATTRIBUTE feat_arab NUMERIC
@ATTRIBUTE feat_polityk NUMERIC
@ATTRIBUTE feat_jordański NUMERIC
@ATTRIBUTE feat_jordanii NUMERIC
@ATTRIBUTE feat_saudyjski NUMERIC
@ATTRIBUTE feat_libański NUMERIC
@ATTRIBUTE feat_palestyńskiej NUMERIC
@ATTRIBUTE feat_algierii NUMERIC
@ATTRIBUTE feat_algierski NUMERIC
@ATTRIBUTE feat_ammanie NUMERIC
```

Poniżej załączono zrzut ekranu przedstawiający fragment przykładowego, wygenerowanego pliku ARFF ukazujący początek sekcji definiującej osobników (tablice rzadkie dokumentów):

```
341 @ATTRIBUTE feat_portalu NUMERIC
342 @ATTRIBUTE feat_ulicy NUMERIC
343
344 @data
345 {0 Albania}
346 {0 Albania, 1 1, 180 2, 185 2, 282 2}
347 {0 Albania, 282 1}
348 {0 Albania, 1 1, 104 1, 246 1, 282 1}
349 {0 Albania, 1 1, 104 1, 246 1, 282 1}
350 {0 Albania, 1 1, 104 1, 246 1, 282 1}
351 {0 Albania, 1 1, 104 1, 246 1, 282 1}
352 {0 Albania, 1 1, 104 1, 106 1, 246 1, 282 1}
353 {0 Albania, 5 2, 14 1, 36 1, 110 1, 237 1, 261 1}
354 {0 Albania, 1 1, 3 1, 5 1, 7 1, 9 1, 15 1, 75 1, 288 1}
355 {0 Albania, 2 1, 7 1, 9 1}
356 {0 Albania, 2 1, 115 1, 174 1}
357 {0 Albania, 2 1, 4 2}
358 {0 Albania, 1 1, 4 1, 7 1, 11 1, 18 1, 19 1, 63 1, 124 1}
359 {0 Albania, 2 1, 4 1, 5 1, 174 2, 201 1}
360 {0 Albania, 1 3, 2 1, 3 1, 7 1, 175 1, 176 1, 181 1, 303 5}
361 {0 Albania, 1 1, 2 1, 3 2, 12 1, 18 1, 19 2}
362 {0 Albania, 3 2, 4 2, 7 1, 174 1}
363 {0 Albania, 1 2, 2 2, 3 1, 4 2, 5 2}
364 {0 Albania, 2 2, 4 1, 174 1}
365 {0 Albania, 2 1, 3 3, 42 1, 201 2, 203 1, 237 1, 281 1, 284 1, 289 1}
366 {0 Albania, 9 1, 14 1, 32 1}
367 {0 Albania, 1 2, 2 1, 5 2, 22 1, 174 1, 200 1, 201 1, 207 1, 236 1}
368 {0 Albania, 22 1, 241 2, 314 1}
369 {0 Albania, 1 1, 3 2, 4 1, 5 1, 7 1, 131 1, 241 1}
370 {0 Albania, 2 1, 3 4, 4 1, 12 1, 131 1}
371 {0 Albania, 1 1, 2 1, 5 1, 9 2, 201 1, 207 2}
372 {0 Albania, 1 2, 7 1, 203 1, 241 3}
373 {0 Albania, 2 1, 9 1, 11 1, 85 1}
374 {0 Albania, 1 1, 123 1}
375 {0 Albania, 3 2, 4 1, 7 1}
376 {0 Albania, 2 1, 3 2, 83 1}
377 {0 Albania, 2 1, 3 2, 295 1}
378 {0 Albania, 1 2}
379 {0 Albania, 1 1, 2 1, 3 1, 5 1, 9 1, 203 1}
380 {0 Albania, 2 1, 9 1, 201 1, 207 1}
381 {0 Albania, 2 1, 9 1, 15 1, 201 1, 207 1}
382 {0 Albania, 2 1, 3 2}
383 {0 Albania, 2 1, 5 1, 9 1, 78 1}
384 {0 Albania, 1 1, 2 1, 9 1, 162 1, 169 1}
385 {0 Albania, 2 1, 9 1, 11 1}
386 {0 Albania, 9 1, 78 1, 288 1, 289 1, 295 1}
387 {0 Albania, 1 1, 2 1, 3 1, 5 1, 9 1}
388 {0 Albania, 3 4, 4 1, 7 1, 137 1, 201 1, 207 1}
389 {0 Albania, 2 1, 9 1, 201 1, 207 1}
390 {0 Albania, 2 1, 9 1, 201 1}
391 {0 Albania, 2 1, 9 1}
392 {0 Albania, 2 1, 9 1, 201 1, 289 1}
393 {0 Albania, 1 1, 2 1, 9 1}
394 {0 Albania, 2 1, 9 1, 49 1}
395 {0 Albania, 2 1, 9 1}
396 {0 Albania, 2 1, 4 1, 11 1, 12 1, 18 1, 19 2, 174 1}
```

3. Badania - wstęp

Wszystkie badania wykonano korzystając ze środowiska do uczenia maszynowego WEKA stworzonego na Uniwersytecie Waikato w Nowej Zelandii. Narzędzie to oferuje ogromną ilość systemów do nauki i badań uczenia maszynowego, w tym metod klasyfikacji takich jak drzewa decyzyjne czy klasyfikatory bayesowskie.

Dla każdego z badań w celu podziału zbioru danych na dane treningowe i dane testowe użyto 10-krotnej walidacji, z wyjątkiem badań w których opisie została sprecyzowana inna metodyka.

Metodyka K-krotnej walidacji polega na podziale zbioru danych na K zbiorów (folds). Następnie kolejno każdy z nich użyty jest jako zbiór testowy, a suma pozostałych jako zbiór treningowy. Analiza wykonywana jest więc K razy, a wyniki ostateczne są uśredniane. W ten sposób uzyskujemy możliwość powtarzalności eksperymentu (a w związku z tym weryfikacji wyników) nie poświęcając dużej ilości danych na dane testowe (zachowując jak najwyższą ilość danych treningowych).

W celu wykonania odpowiednich badań przygotowano wcześniej (programem opisanym w punkcie drugim sprawozdania) pliki danych o różnej ilości ekstrahowanych cech:

- features_169.arff – 169 cech
- features_335.arff – 335 cech
- features_501.arff – 501 cech
- features_667.arff – 667 cech
- features_983.arff – 983 cechy

Poza liczbą cech opisujących dokumenty zestaw danych wzorcowych w każdym pliku jest identyczny.

Poniżej załączono poglądowy zrzut ekranu przykładowego podsumowania wyników wygenerowanego przez środowisko WEKA dla jednego z algorytmów:

```
Time taken to build model: 0.4 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        6270               63.7389 %
Incorrectly Classified Instances      3567               36.2611 %
Kappa statistic                          0.6265
Mean absolute error                      0.0226
Root mean squared error                  0.1181
Relative absolute error                 39.6016 %
Root relative squared error             69.8952 %
Total Number of Instances             9837
```

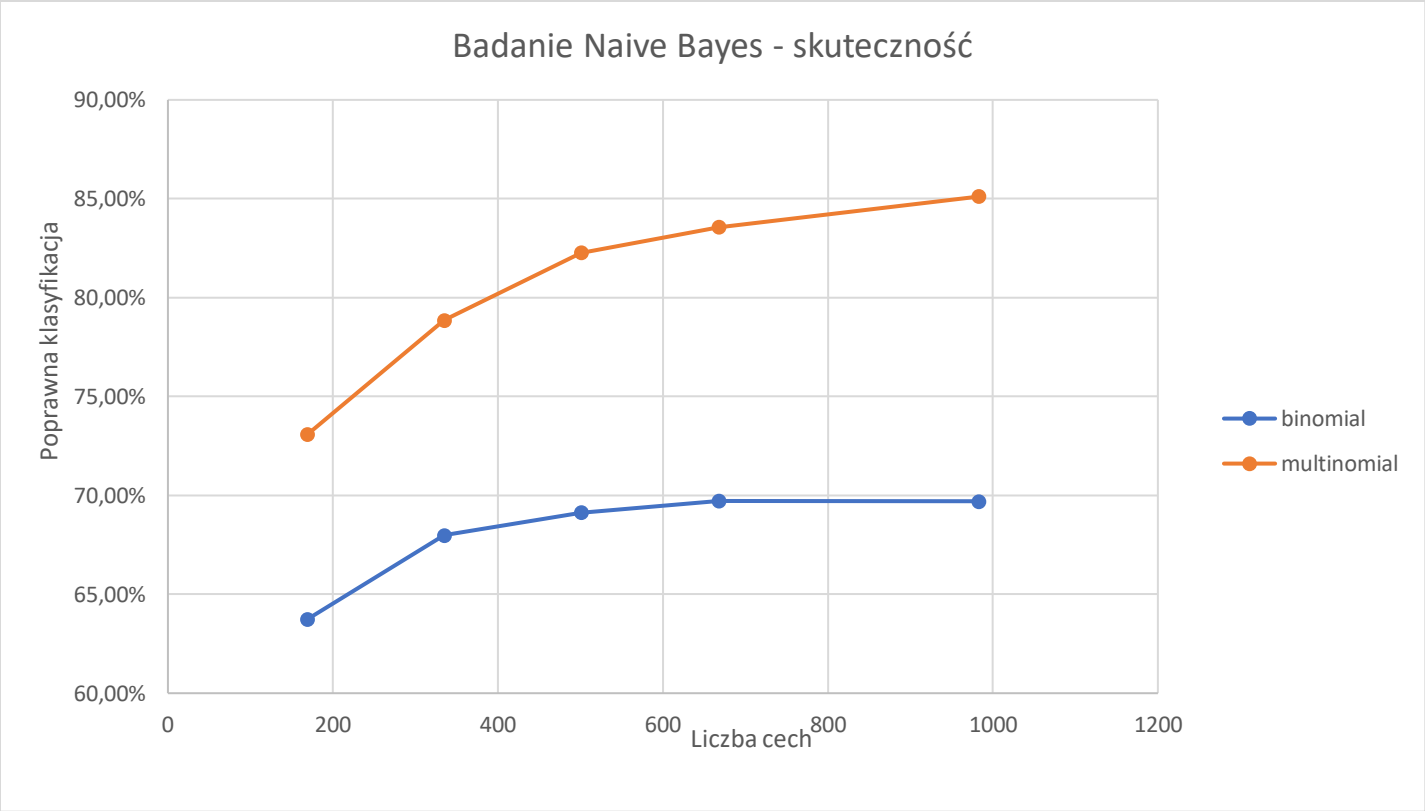
4. Badania – Naive Bayes

a) Opis wariantów „binomial” i „multinomial” algorytmu:

Algorytm Naive Bayes w wersji „binomial” rozpatruje tylko i wyłącznie fakt występowania danej cechy w dokumencie. Algorytm w wersji „multinomial” jest jego usprawnieniem uwzględniającym także ich liczność.

b) Badanie porównujące skuteczności oraz czasy uczenia dla algorytmu w wersji „binomial” i „multinomial” w zależności od ilości wyselekcjonowanych cech.

Poniżej załączono wykres zależności skuteczności algorytmów (wyrażonej w stosunku poprawnie zaklasyfikowanych do błędnie zaklasyfikowanych instancji) od ilości wyekstrahowanych cech:



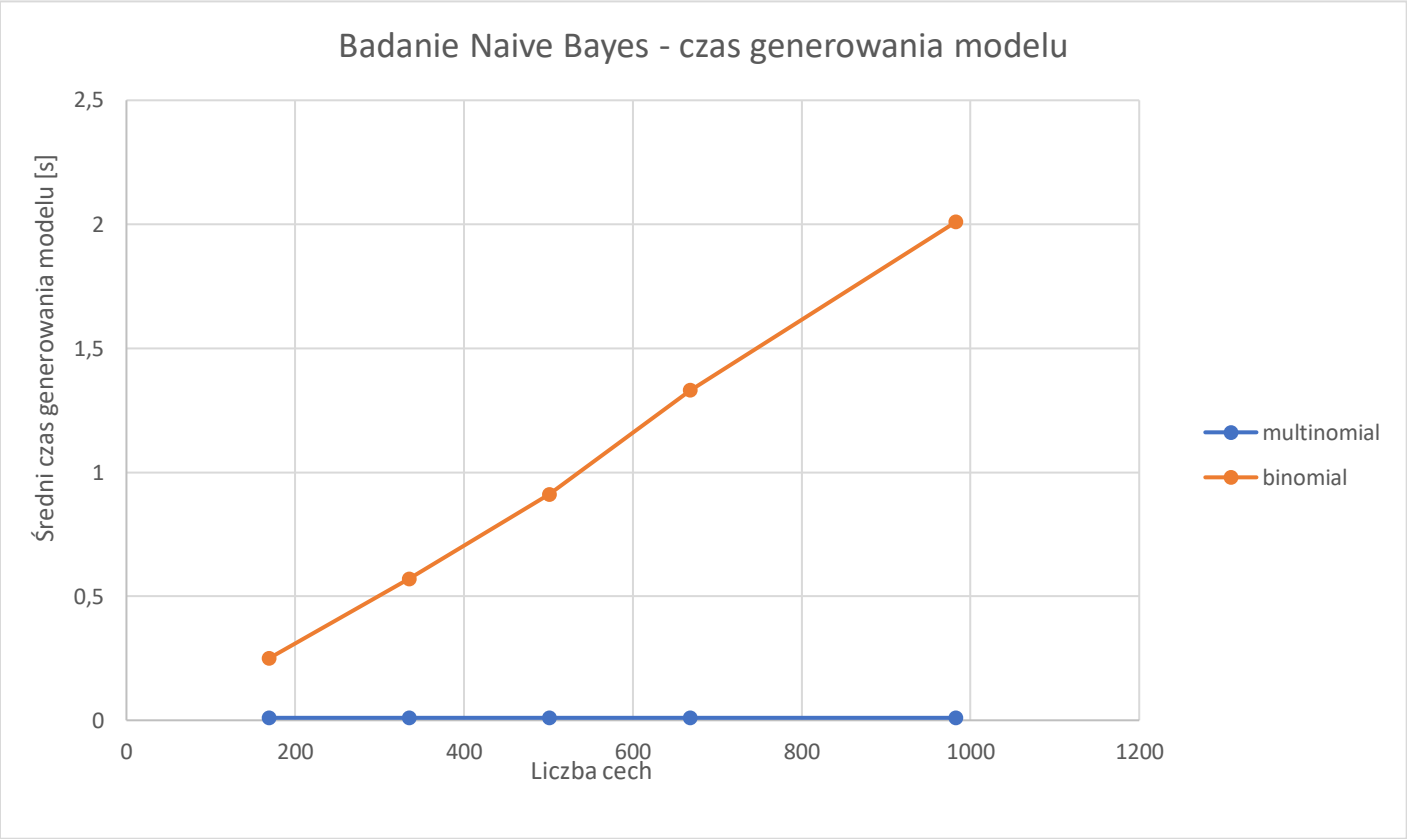
Wnioski:

- zgodnie z intuicją uwzględnienie liczności wystąpień słów w dokumencie znacznie zwiększa skuteczność algorytmu (przewaga algorytmu w wersji „multinomial”)

- im większa liczba ekstrahowanych cech tym większa skuteczność algorytmów w obu formach. Powyżej pewnego progu zysk staje się jednak coraz mniejszy. Fakt ten może wynikać z tego, że od pewnego momentu wszystkie znacząco relewantne cechy zostają zawarte w definicjach instancji.

c) Badanie porównujące czasy uczenia dla algorytmu w wersji „binomial” i „multinomial” w zależności od ilości wyselekcjonowanych cech.

Poniżej załączono wykres zależności czasu potrzebnego do utworzenia modelu dla algorytmów w obu wariantach (wyrażonego w sekundach) od ilości wyekstrahowanych cech:



Wnioski:

- wbrew intuicji generowanie modelu dla algorytmu w wersji uwzględniającej liczności występowania cech w dokumencie jest znacznie szybsze od standardowej metody „binomial”. Fakt ten wynika z tego, że algorytm w tej wersji nie potrzebuje procesować cech nie występujących w dokumentach. Do obliczeń brane są pod uwagę tylko liczności cech większe od zera. W przypadku problemu klasyfikacji dokumentów obiekty definiowane są dużą liczbą cech, jednak tylko nieliczne z nich występują w dokumencie (dlatego idealną reprezentacją są matryce rzadkie) i tym większy jest zysk tego wariantu algorytmu.

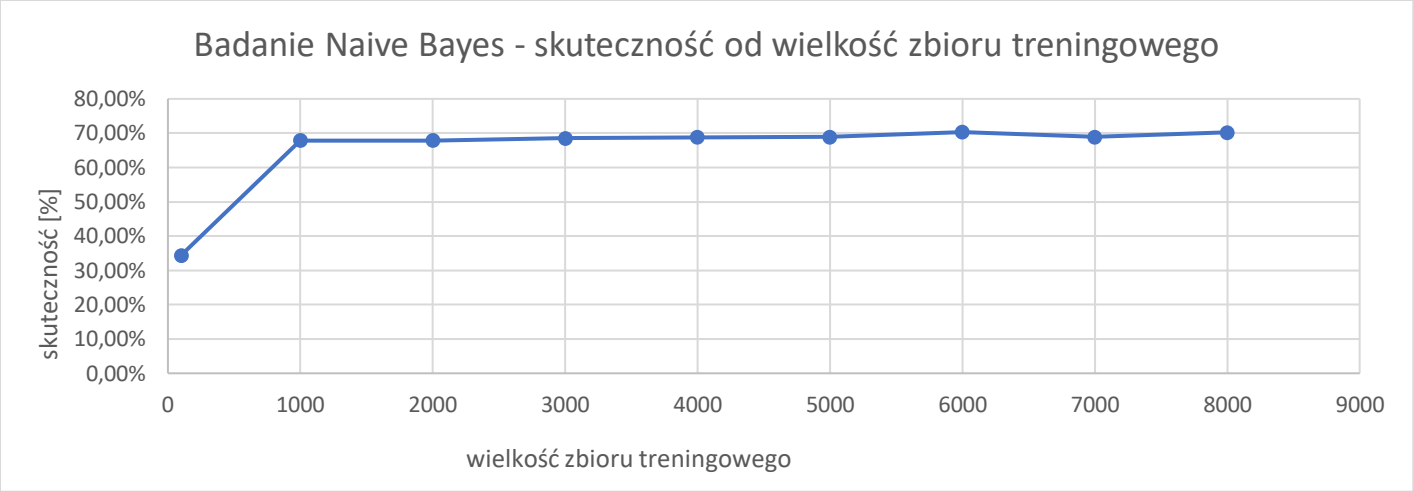
- ponadto dla algorytmu „binomial” zauważono dodatkowo, że czas tworzenia modelu jest silnie zależny od ilości cech. Powód jest analogiczny. Ten wariant algorytmu musi wziąć pod uwagę w obliczeniach wszystkie cechy dla dokumentu niezależnie od tego czy w nim występują czy nie. Dla algorytmu „multinomial” wzrost ten jest dużo mniejszy (niezauważalny w powyższym przykładzie), ponieważ przy wzroście liczby definiowanych cech tylko nieznacznie wzrasta liczba cech występujących w danym dokumencie.

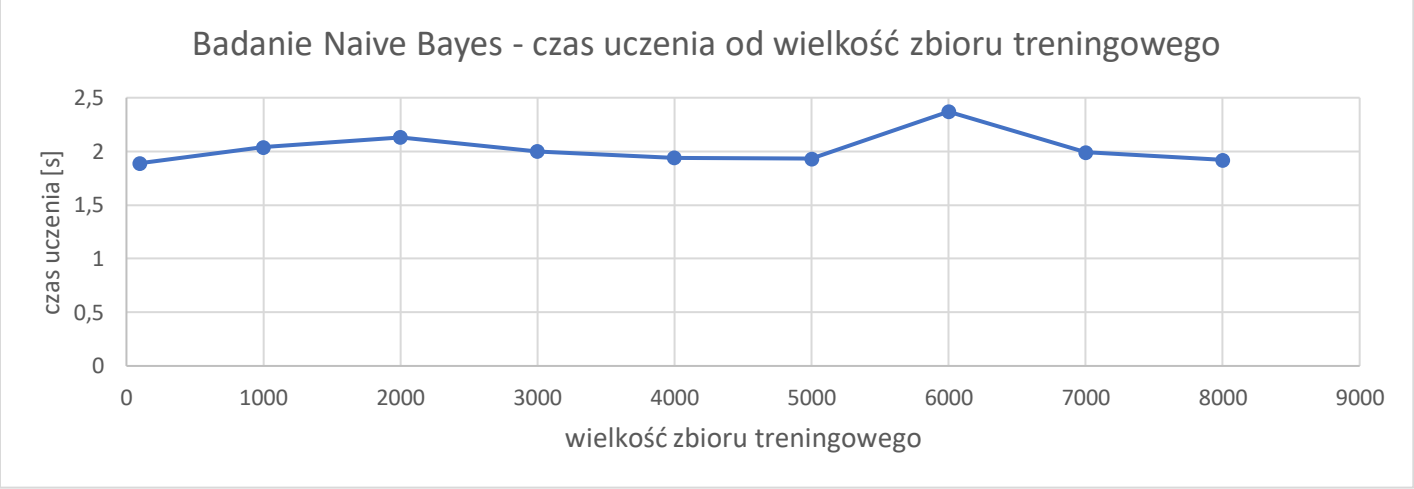
d) Badanie czasu i skuteczności algorytmu Naive Bayes w wariancie „binomial” w zależności od wielkości zbioru treningowego.

- w celu wykonania tego badania zrezygnowano z metodyki k-krotnej walidacji, aby umożliwić łatwiejszą manipulację wielkością zbioru treningowego

- skorzystano z pliku o największej liczbie wyekstrahowanych cech – 983

Poniżej zamieszczono wykresy dokumentujące wykonane badanie:





Wnioski:

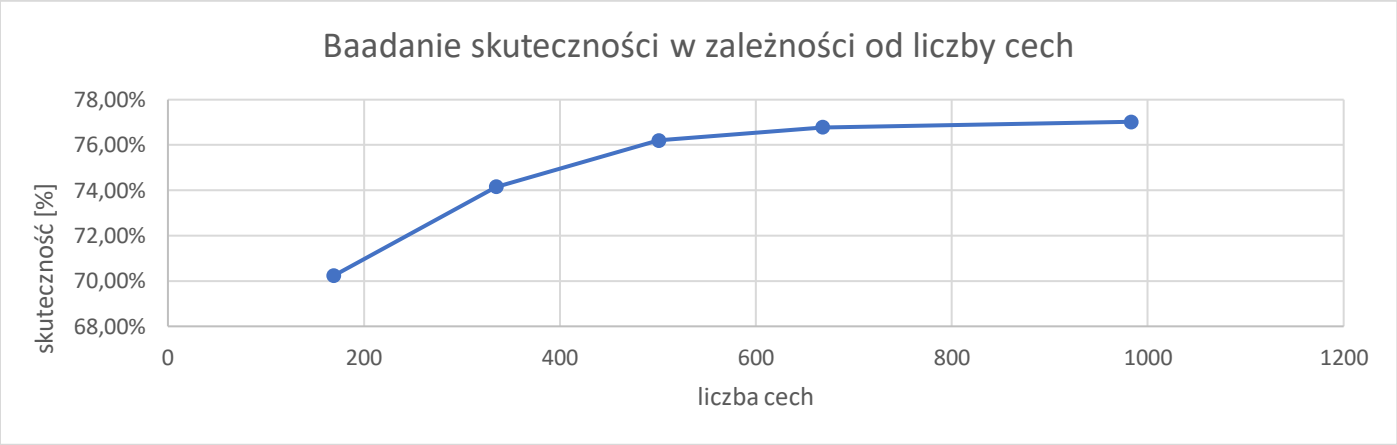
- badanie nie wykazało znaczącej zależności pomiędzy wielkością zbioru a czasem uczenia modelu czy skutecznością otrzymywanych wyników (poza granicznym wypadkiem, gdzie dobrano bardzo mały zbiór 100 instancji)

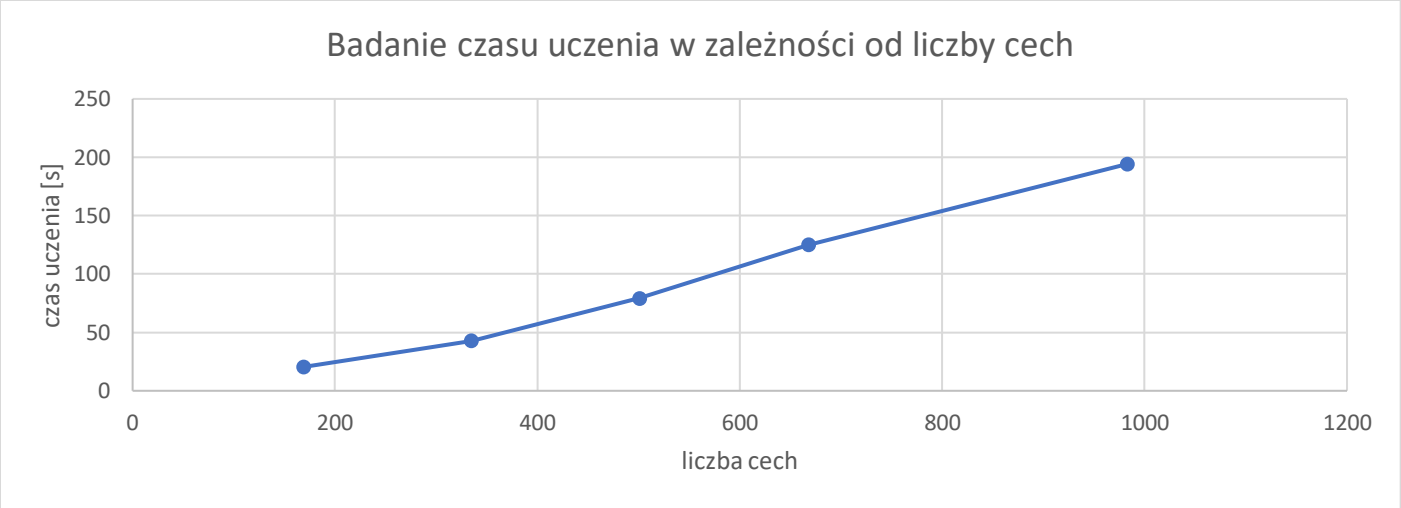
- czynnikiem tłumaczącym ten efekt może być fakt, że cechy relewantne były selekcjonowane na podstawie całego zbioru 10000 dokumentów (niezależnie od dobranej potem wielkości zbioru treningowego). Ponadto należy zauważyć, że algorytm z całego zbioru danych dobiera do zbioru uczącego równo rozdystrybuowane dokumenty reprezentatywne dla każdej z klas, co może znacząco pomagać w zachowaniu efektywności przy zmniejszaniu zbioru uczącego.

5. Badania – drzewa decyzyjne C4.5

Drzewa decyzyjne są stosunkowo starym rozwiązaniem i nie są pierwszym wyborem w rozwiązywaniu problemu klasyfikacji tekstu. Ich skuteczność i czas uczenia jest jednak silnie zależny od jakości selekcji cech, więc są one dobrym sposobem na przetestowanie tego etapu. Do badań wybrano drzewo typu C4.5, które jest rozwinięciem algorytmów klasycznego drzewa ID3.

a) Poniżej załączono wykresy dokumentujące badanie czasu i skuteczności drzewa decyzyjnego w zależności od liczby selekcjonowanych cech:





Wnioski:

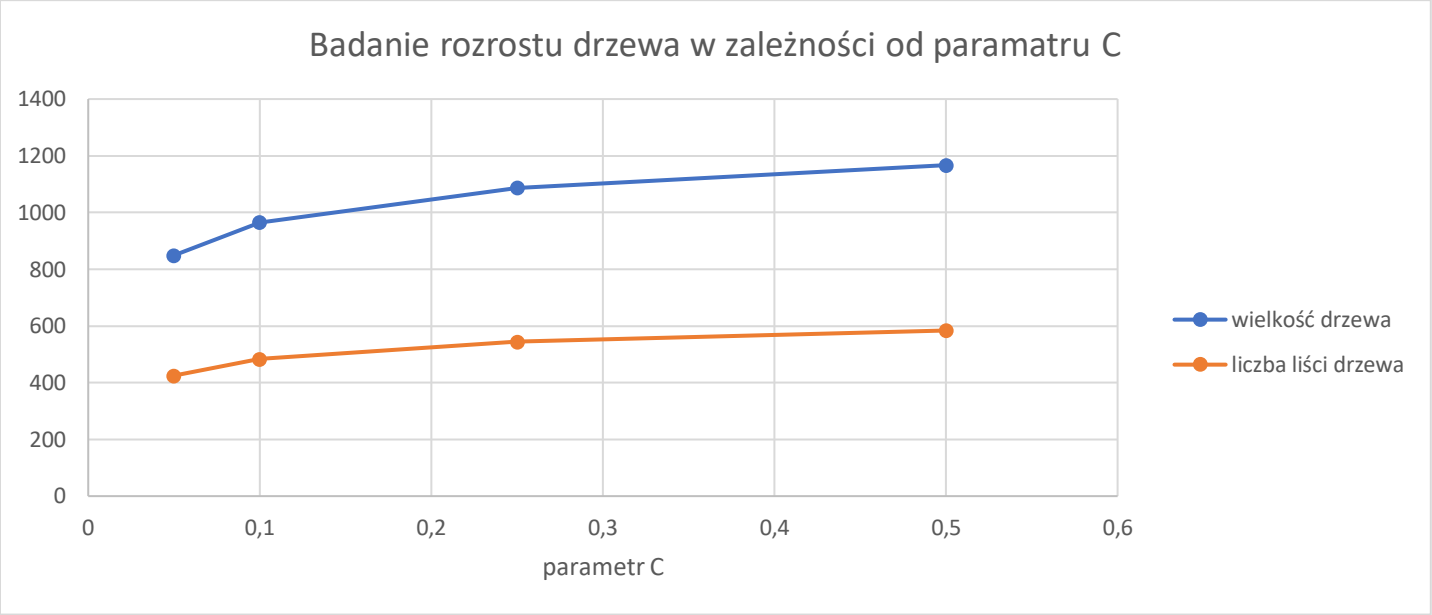
- zgodnie z teorią potwierdza się silna zależność pomiędzy czasem uczenia, a liczbą cech. Większa liczba cech wpływa bezpośrednio na wielkość budowanego drzewa, liczbę rozgałęzień i złożoność procesu wycinania niepotrzebnych gałęzi

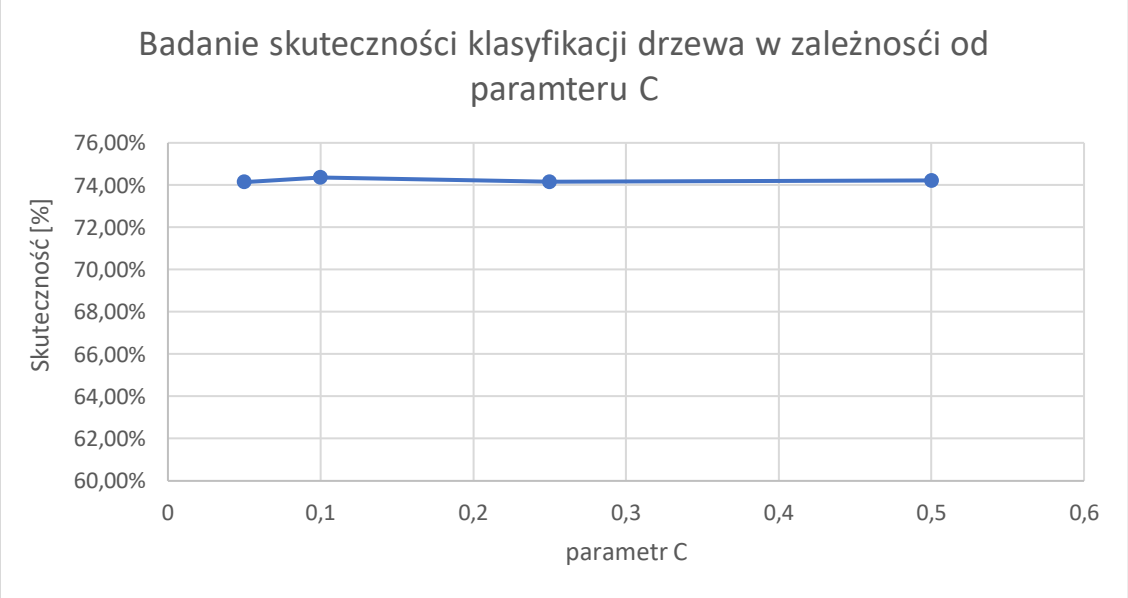
- należy jednak zauważyć, że mimo powyżej opisanej wady, algorytm osiąga znacznie lepsze rezultaty dla większej liczby selekcjonowanych cech

- powyżej opisana tendencja maleje jednak po przekroczeniu pewnego progu i algorytmowi nie udaje się osiągnąć skuteczności na poziomie skuteczności algorytmów bayesowskich

b) Poniżej załączono wykresy dokumentujące badanie skuteczności drzewa decyzyjnego oraz rozrost drzewa w zależności od progu decyzyjnego C używanego przy podejmowaniu decyzji o wycięciu danej podgałęzi drzewa. Badanie przeprowadzono dla pliku o liczbie cech równej 335.

Poniżej załączono wykresy dokumentujące rezultaty badania:





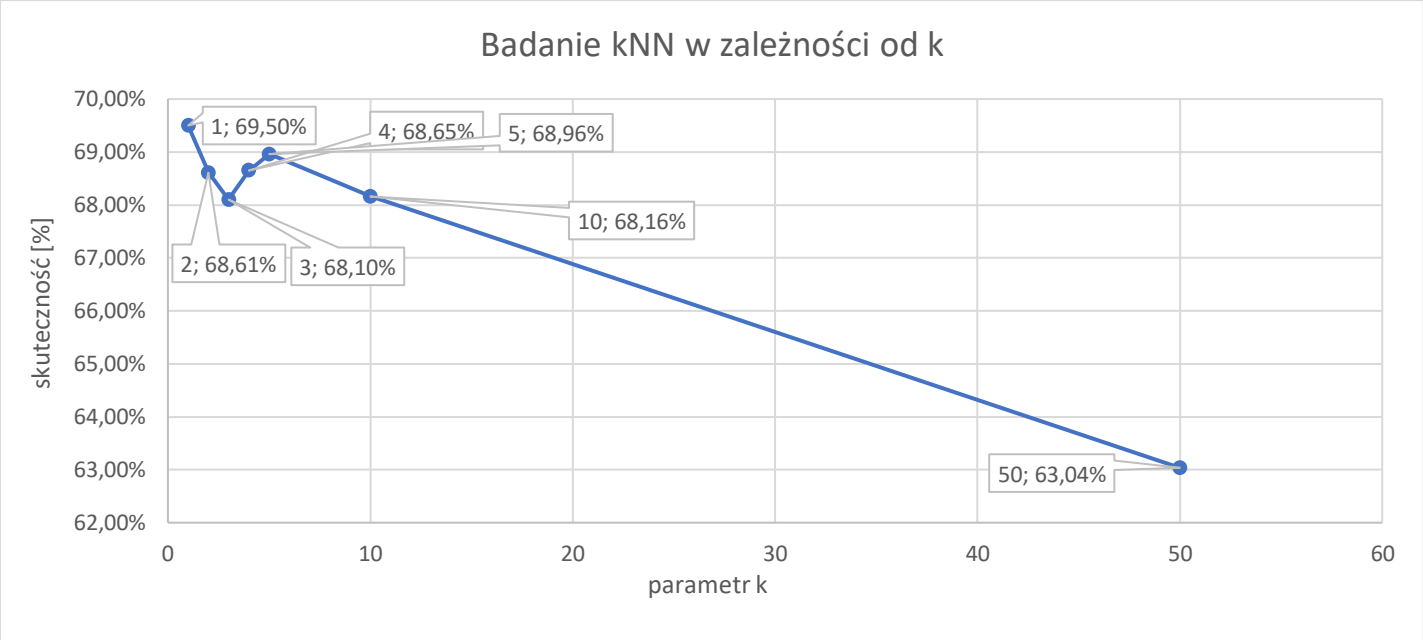
Wnioski:

- nawet przy stosunkowo niskim progu ufności C (odrzucanie dużej ilości gałęzi) udaje się zachować maksymalną skuteczność klasyfikacji. Zyskiem tej konfiguracji jest skrócenie czasu klasyfikacji (mniejszy rozrost drzewa)

6. Badanie dodatkowe – algorytm kNN (k-Nearest Neighbours)

Cechą szczególną wybranego algorytmu jest to, że nie potrzebna jest budowa modelu. Ewaluacja odbywa się poprzez odniesienie do liczby k najbliższych sąsiadów (osobników najbardziej zbliżonych cechami). Pomimo tego, że zaoszczędzony zostaje czas na generowanie modelu, stosunkowo czasochłonny staje się etap ewaluacji.

Zbadano skuteczność algorytmu w odniesieniu do hiperparametru k algorytmu i załączono wykres dokumentujący badanie:



Wnioski:

Z badania wynika, że najlepszą wartością k dla omawianego zbioru wzorcowego jest k = 1. Oznacza to, że zawsze najtrafniejszą decyzją jest sugerowanie się dokumentem o najbardziej zbliżonych cechach. Im większy parametr k, tym większy staje się szum w procesie ewaluacji. Potwierdza to dobrą metodykę selekcji cech relewantnych dla poszczególnych klas.

Z drugiej strony należy zauważyć, że maksymalna osiągana skuteczność algorytmu jest stosunkowo niska i nie dorównuje efektom uzyskiwanym klasyfikatorami bayesowskimi.

7. Najlepszy rezultat

W celu uzyskania jak najlepszej skuteczności postanowiono na podstawie powyższych badań skorzystać z klasyfikatora bayesowskiego w wersji „multinomial”. Programem opisanym w punkcie drugim sprawozdania wygenerowano dodatkowy plik „features_1301.arff” zawierający dokumenty definiowane liczbą cech równą 1301.

Tym sposobem udało się uzyskać średnią skuteczność nieznacznie większą od poprzednio uzyskanej tym klasyfikatorem (dla pliku zawierającego 986 cechy):

Średnia skuteczność dla 986 cech: 85,11%
Średnia skuteczność dla 1301 cech: 85,82%

Oznacza to, że potencjał dalszej selekcji większej ilości cech dla tego klasyfikatora wyczerpał się i nie przyniesie kolejnych, lepszych rezultatów.

Poniżej załączono zrzut ekranu wyników przeprowadzonego eksperymentu:

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8442               85.8188 %
Incorrectly Classified Instances      1395               14.1812 %
Kappa statistic                          0.8538
Mean absolute error                      0.0089
Root mean squared error                  0.0835
Relative absolute error                 15.5995 %
Root relative squared error             49.4488 %
Total Number of Instances             9837
```

8. Podsumowanie

Udało się zaimplementować skuteczny program w języku C# dokonujący selekcji N*ilość klas cech na podstawie zbioru wzorcowego. Selekcja opiera się na algorytmie Mutual Information, a plikiem wynikowym programu jest plik w formacie „arff” zawierający wyekstrahowane cechy dla każdego dokumentu ze zbioru wzorcowego.

Zbadano skuteczność drzew decyzyjnych i klasyfikatorów bayesowskich (oraz dodatkowo k-NN) w rozwiązywaniu opisywanego problemu klasyfikacji korzystając ze środowiska WEKA. Najlepszym rozwiązaniem okazał się klasyfikator bayesowski w wersji „multinomial” dla liczby wyselekcjonowanych cech równej 1301. Dodatkowo jest on także najszybszym rozwiązaniem z prawie zerowym czasem uczenia się i klasyfikowania danych. Najlepsza skuteczność wyniosła 85.82% poprawnie zaklasyfikowanych dokumentów.