Politechnika Wrocławska

Indukcyjne Metody Analizy Danych

Ćwiczenie 1 - Naive Bayes

Marcel Cielinski
Index: 236747

prowadzący
dr inż. Paweł Myszkowski

2 kwietnia 2020

Wprowadzenie

1.1 Problem

Celem ćwiczenia było poznanie naiwnego klasyfikatora probabilistycznego, opartego na twierdzeniu Bayesa. Do składowych zadania zalicza się samodzielną implementację w języku *Python* oraz przeprowadzenie badań na trzech, ustalonych na potrzeby ćwiczenia, zbiorach danych. Należało przetestować wpływ stosowania wygładzenia danych, wybranych metod dyskretyzacji i użycia walidacji krzyżowej do oceny skuteczności algorytmu poprzez obserwację metryk mówiących o jakości klasyfikatora.

1.2 Naiwny klasyfikator Bayesowski

Formę uczenia nadzorowanego dzielimy na zadania regresji (gdy modelowane są wartości o ciągłym charakterze danych) oraz na zadania klasyfikacji (gdzie zmienne wyjściowe są klasami). Opisywany algorytm jest przykładem tego drugiego typu i jako, że zawiera się w grupie uczenia z nadzorem, każda z instancji danych wejściowych ma przypisaną właściwą etykietę. Celem stosowania takiego narzędzia jest wyszkolenie go na próbce danych, które posiadamy, a następnie (z pewną dozą pewności) przyporządkowywanie etykiet dla nowych wektorów wejściowych.

Opisywany klasyfikator opiera się na na twierdzeniu Bayesa (Wzór 1, gdzie X to wektor wejściowy, a Y to klasa) o prawdopodobieństwie warunkowym oraz na założeniu o wzajemnej niezależności atrybutów danych wejściowych. Jako, że założenie o niezależności zwykle nie pokrywa się z rzeczywistością, algorytm nazywany jest „naiwnym". Mimo to oraz mimo faktu, iż jest to jedno z prostszych narzędzi ML, naiwny klasyfikator Bayesowski często daje dobre wyniki, co zaobserwować będzie można w opisywanym ćwiczeniu.

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{1}$$

Działanie algorytmu oparte jest o prawdopodobieństwie *a priori* przynależności do klas względem atrybutów. Oznacza to nic innego jak sugerowanie się zależnościami odkrytymi w fazie treningowej, w celu wnioskowania o przynależności klasowej nowego badanego obiektu. Jednym z problemów, które mają wpływ na działanie takiego schematu dopasowywania, jest trudność związana z atrybutami reprezentowanymi przez wartości ciągłe. Rozwiązaniem w tym przypadku może być dyskretyzacja danych, a jej wpływ zostanie opisany niżej. Inną przeszkodą może okazać się brak wygładzenia danych, co także będzie przedmiotem rozważań.

Implementacja

Do implementacji naiwnego klasyfikatora Bayesa użyto biblioteki uczenia maszynowego *sklearn*. Wykorzystano modele `GaussianNB` oraz `MultinomialNB`. Oba zakładają, że dane pochodzą z rozkładu (odpowiednio Gaussa i wielomianowego). Na podstawie danych treningowych obliczane są ich parametry. W kodzie wykorzystano także takie bibliotegi jak: *pandas*, *matplotlib*, *numpy* oraz *seaborn*.

2.1 Ocena jakości

Do oceny jakości klasyfikatora stosuje się metryki związane, bądź wynikające z macierzy błędów (ang. Confusion matrix). Po wykonaniu predykcji dopasowania etykiet (klas), możliwe jest zbadanie prawidłowości dopasowań. Właśnie w tym celu posługujemy się tablicą pomyłek, która zestawia liczebność instancji prawidłowo i nieprawidłowo oznaczonych przez klasyfikator. Podstawowe takie metryki to:

- *Accuracy* - dokładność - jak dobre są ogólne predykcje klasyfikatora. Jest to stosunek dobrze oznaczonych klas do wszystkich oznaczeń.
- *Precission* - precyzja - mówi o precyzyjności klasyfikatora. Jest to zdolność klasyfikatora do nie oznaczania negatywnych próbek jako pozytywne.
- *Recall* - czułość - jak dobrze klasyfikator radzi sobie ze znajdywaniem pozytywnych próbek. Jest to stosunek próbek dobrze zaklasyfikowanych jako pozytywne przez wszystkie rzeczywiście pozytywne.
- *F1-score* - oznaczane jako FSC - jest to w istocie średnia harmoniczna czułości oraz precyzji.

Do badania większości testów posłuży głownie miara FSC. Jest to często wykorzystywana miara, która pozwoli na wymiernie dobre porównanie różnych podejść. W implementacji wykorzystany został pakiet *sklearn.metrics*, który udostępnia wszystkie powyżej wymienione metryki.

2.2 Wygładzanie

Wygładzenie danych polega na zwiększeniu częstości występowania danego atrybutu. Dzięki temu można wykluczyć zerowe prawdopodobieństwa ich pojawienia się, co jest bezpośrednim powodem istoty stosowania wygładzenia w fazie przygotowywania danych. Zastosowane w ćwiczeniu modele (`GaussianNB` oraz `MultinomialNB`) domyślnie korzystają z tego narzędzia. W celach porównawczych, dla przeprowadzanych badań, te domyślne ustawienia były także dezaktywowane. Jest możliwe poprzez odpowiednie ustawianie parametrów (odpowiednio *alpha* i *var_smoothing*).

2.3 Dyskretyzacja

Stosując naiwny klasyfikator Bayesa, w pewnym sensie utrudnieniem są atrybuty o wartościach ciągłych. Metodą, jaką można się wesprzeć, jest dyskretyzacja. Jak sama nazwa wskazuje, polega na zamianie danych ciągłych na dyskretne. Zwykle dzieje się to na zasadzie tzw. kubełkowania. W skrócie działanie jest następujące: Na początku tworzona jest odpowiednia ilość kubełków, z których każdy będzie etykietą dla określonego zakresu wartości $([attr\_val_i, attr\_val_{i+1}), [attr\_val_{i+1}, attr\_val_{i+2}), \ldots)$. Kubełki muszą zamykać całą przestrzeń przyjmowanych przez atrybuty wielkości. Następnie wartości atrybutów każdej instancji muszą zostać przypisane do odpowiadających im kubełków. Na tym etapie procedura jest zakończona. Oczywiście wraz z takim przekształceniem zmiennych, zmianie ulega także ich rozkład. Jest to wprawdzie uproszczenie danych kosztem pewnej utraty informacji.

Na potrzeby realizacji zadania, skorzystano z modułu `KBinsDiscretizer` pakietu *sklearn.preprocessing*. Można go parametryzować poprzez ustawianie ilości kubełków (*n_bins*) oraz strategii przyporządkowywania wartości odpowiednim kubełkom. Obiektem badań były trzy strategie:

- *uniform* - gdzie każdy kubełek jest równej wielkości przedziału.
- *quantile* - gdzie każdy kubełek ma taką samą ilość instancji.
- *kmeans* - gdzie próbki w ramach jednego kubełka miały wspólny najbliższy środek klastra.

2.4 Kroswalidacja

Kroswalidacja (sprawdzian krzyżowy) jest stosowana w celu lepszej oceny działania modeli, takich jak naiwny klasyfikator Bayesa. Procedura determinuje podział danych na treningowe i testowe. A jej wykorzystanie tłumaczy się unikaniem overfittingu.

W realizacja zadania wykorzystano dwie metody, dostępne w pakiecie *sklearn.model_selection*. Są nimi:

- `KFold` - kroswalidacja - gdzie całkowity zbiór danych jest dzielony na K równolicznych podzbiorów (parametr *n_splits*), z których kolejno każdy jest traktowany jako zbiór testowy, kiedy model trenowany jest na połączonych (*K-1*) pozostałych podzbiorach. Otrzymuje się wówczas K wyników, z których następnie liczona jest średnia.
- `StratifiedKFold` - kroswalidacja stratyfikowana - jej działanie jest analogiczne do powyżej opisanego, wariantu zwykłego. Różnica polega na tym, że całkowity zbiór jest dzielony (na K części, także parametrem *n_splits*) w miarę możliwości na podzbiory o proporcjonalnym rozkładzie klas - zgodnie z proporcją istniejącą w całościowym zbiorze.

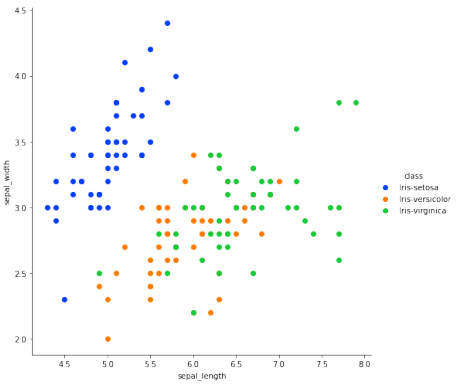
W badaniach ustawiano parametr *shuffle* = *True* w celu zapewnienia wstępnego losowego rozłożenia danych (często zbiory danych wejściowych są sortowane po klasach). Testowany był także wpływ ilości foldów (K) na otrzymywane wyniki, dla obu metod.

2.5 Zbiory danych

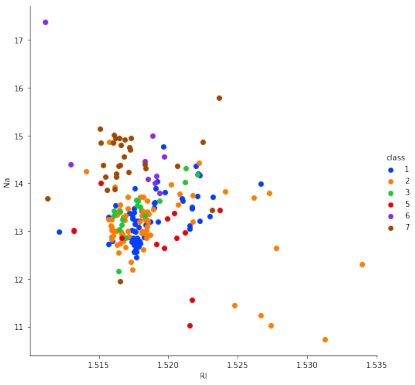
Do walidacji zaimplementowanych funkcjonalności wykorzystano łącznie cztery zbiory danych. Jeden w celu sprawdzenia działania i trzy pozostałe w celach testów i przeprowadzenia badań. Te zbiory, to:

- *iris* - gdzie liczba atrybutów to 4, a klas 3. Zbiór przedstawia odmiany kwiatów Iris, a atrybuty odpowiadają ich cechom.
- *glass* - gdzie liczba atrybutów to 9, a klas 7. Zbiór przedstawia typy szkła, gdzie atrybuty opisują skład chemiczny.
- *wine* - gdzie liczba atrybutów to 13, a klas 3. Zbiór ten zawiera dane o analizie chemicznej win z tego samego regionu, jednak z 3 różnych upraw.
- *seeds* - gdzie liczba atrybutów wynosi 7, a klas 3. Zbiór ten zawiera miary geometrycznych własności 3 różnych odmian ziaren pszenicy.

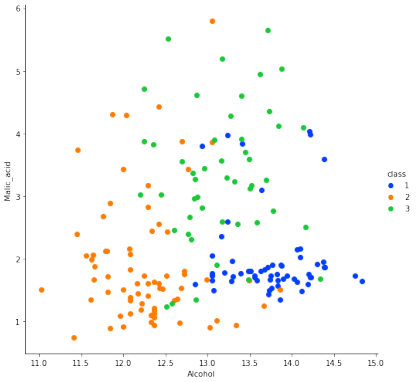
Na Rysunkach 1, 2, 3, 4 przedstawiono przykładowe rozkłady klas względem wybranych atrybutów.



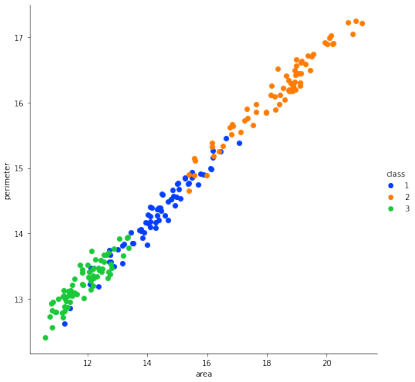
Rysunek 1: Iris



Rysunek 2: Glass



Rysunek 3: Wine



Rysunek 4: Seeds

Wyniki eksperymentów

3.1 Wpływ wygładzenia

Wygładzenie dla zbioru *glass*:

Model	*acc*	*prec*	*rec*	*FSC*
GaussianNB without smoothing	0.0308	0.0051	0.1667	0.0100
GaussianNB with smoothing	0.4615	0.4055	0.4954	0.4097
MultinomialNB without smoothing	0.5538	0.4674	0.4509	0.4400
MultinomialNB with smoothing	0.4769	0.2689	0.3419	0.296

Tablica 1: Wpływ wygładzenia na metryki dla zbioru *glass*

Wygładzenie dla zbioru *wine*:

Model	*acc*	*prec*	*rec*	*FSC*
GaussianNB without smoothing	0.9444	0.9444	0.9545	0.9466
GaussianNB with smoothing	0.9444	0.9444	0.9545	0.9466
MultinomialNB without smoothing	0.8704	0.8682	0.8682	0.8682
MultinomialNB with smoothing	0.8704	0.8617	0.8682	0.8644

Tablica 2: Wpływ wygładzenia na metryki dla zbioru *wine*

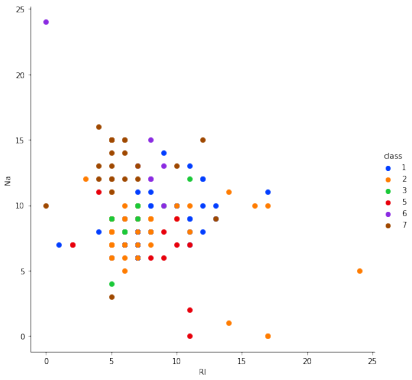
Wygładzenie dla zbioru *seeds*:

Model	*acc*	*prec*	*rec*	*FSC*
GaussianNB without smoothing	0.8889	0.8965	0.9032	0.8973
GaussianNB with smoothing	0.8889	0.8965	0.9032	0.8973
MultinomialNB without smoothing	0.5238	0.6757	0.5964	0.4908
MultinomialNB with smoothing	0.5238	0.6757	0.5964	0.4908

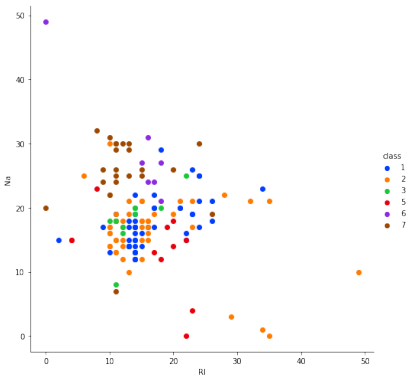
Tablica 3: Wpływ wygładzenia na metryki dla zbioru *seeds*

3.2 Wpływ dyskretyzacji

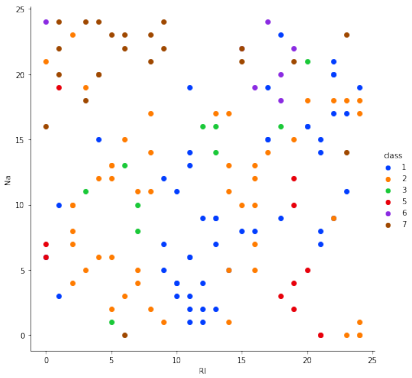
Dyskretyzacja dla zbioru *glass*:



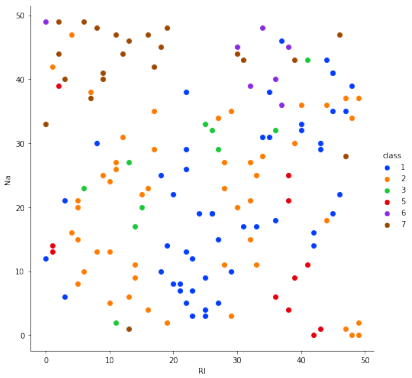
Rysunek 5: *uniform - n_bins = 25*



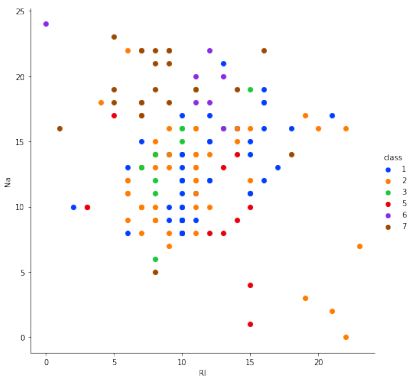
Rysunek 6: *uniform - n_bins = 50*



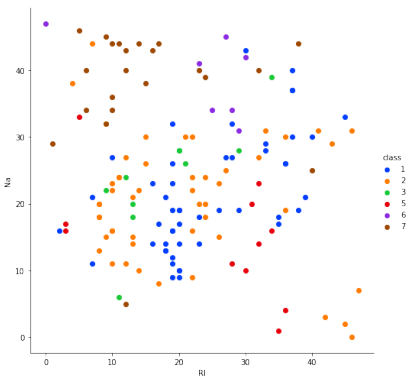
Rysunek 7: *quantile - n_bins = 25*



Rysunek 8: *quantile - n_bins = 50*

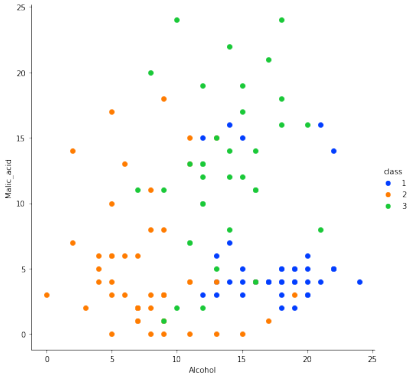


Rysunek 9: *kmeans - n_bins = 25*

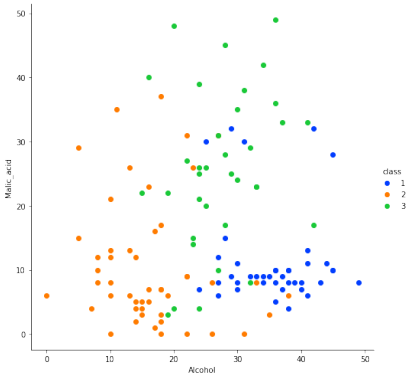


Rysunek 10: *kmeans - n_bins = 50*

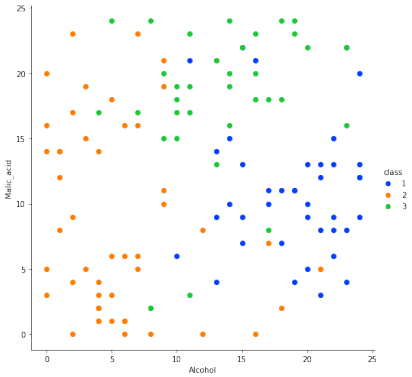
Dyskretyzacja dla zbioru *wine*:



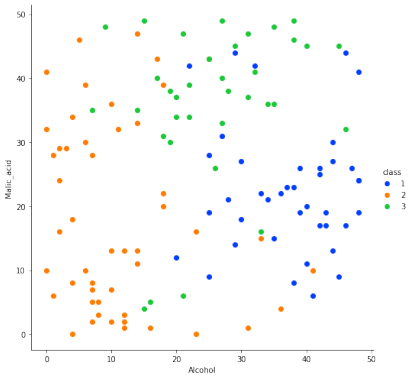
Rysunek 11: *uniform - n_bins = 25*



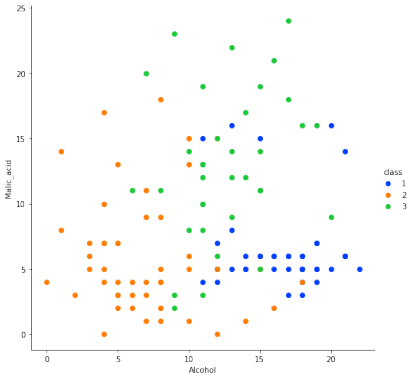
Rysunek 12: *uniform - n_bins = 50*



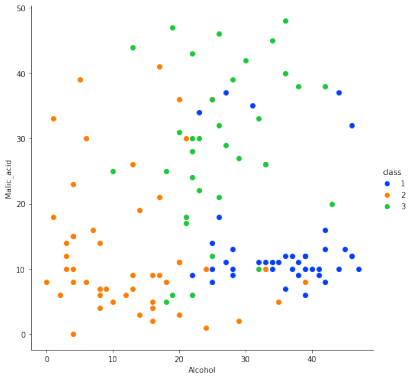
Rysunek 13: *quantile - n_bins = 25*



Rysunek 14: *quantile - n_bins = 50*

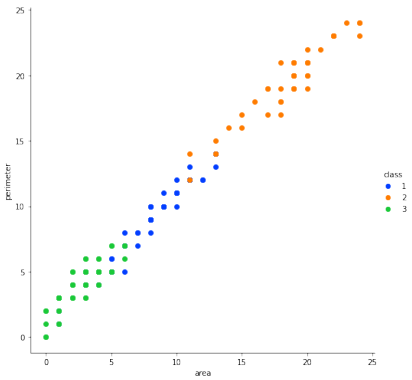


Rysunek 15: *kmeans - n_bins = 25*

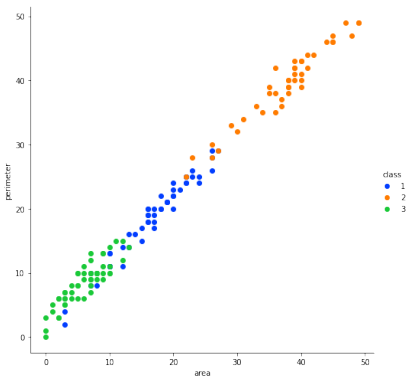


Rysunek 16: *kmeans - n_bins = 50*

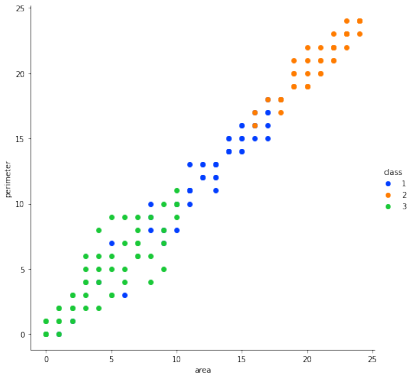
Dyskretyzacja dla zbioru *seeds*:



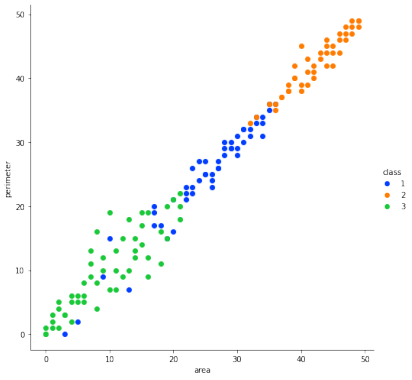
Rysunek 17: *uniform - n_bins = 25*



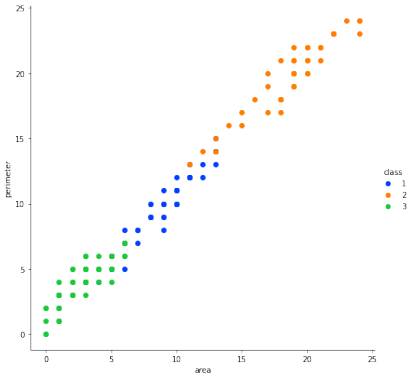
Rysunek 18: *uniform - n_bins = 50*



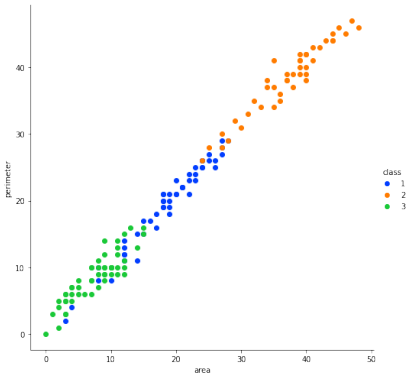
Rysunek 19: *quantile - n_bins = 25*



Rysunek 20: *quantile - n_bins = 50*



Rysunek 21: *kmeans - n_bins = 25*



Rysunek 22: *kmeans - n_bins = 50*

Strategia – kubełki	*glass*	*wine*	*seeds*
bez	0.4400	0.8682	0.4908
uniform – n_bins = 25	0.5448	0.9187	0.8961
uniform – n_bins = 50	0.6037	0.9349	0.8763
quantile – n_bins = 25	0.5080	0.9303	0.8052
quantile – n_bins = 50	0.4979	0.9303	0.8235
kmeans – n_bins = 25	0.6445	0.9349	0.8961
kmeans – n_bins = 50	0.5657	0.9137	0.8763

Tablica 4: Wpływ dyskretyzacji na FSC dla zbiorów: *glass*, *wine*, *seeds*

3.3 Wpływ kroswalidacji

Wpływ kroswalidacji na zbiory *glass*, *wine*, *seeds*:

Model	*glass*	*wine*	*seeds*
Bez kroswalidacji			
GaussianNB without smoothing	0.0100	0.9466	0.8973
GaussianNB with smoothing	0.4097	0.9466	0.8973
MultinomialNB without smoothing	0.4400	0.8682	0.4908
MultinomialNB with smoothing	0.2962	0.8644	0.4908
K = 2			
GaussianNB without smoothing	0.0463	0.9671	0.9048
GaussianNB with smoothing	0.4353	0.9671	0.900
MultinomialNB without smoothing	0.3509	0.8589	0.6071
MultinomialNB with smoothing	0.3622	0.8287	0.777
K = 5			
GaussianNB without smoothing	0.0315	0.9671	0.914
GaussianNB with smoothing	0.4222	0.9671	0.9095
MultinomialNB without smoothing	0.3606	0.8526	0.692
MultinomialNB with smoothing	0.3533	0.8585	0.7621
K = 10			
GaussianNB without smoothing	0.0304	0.9781	0.9042
GaussianNB with smoothing	0.4646	0.9723	0.9046
MultinomialNB without smoothing	0.3253	0.8754	0.7804
MultinomialNB with smoothing	0.3076	0.8477	0.7902

Tablica 5: Wpływ kroswalidacji na FSC dla zbiorów: *glass*, *wine*, *seeds*

Wpływ kroswalidacji stratyfikowanej na zbiory *glass*, *wine*, *seeds*:

Model	*glass*	*wine*	*seeds*
Bez kroswalidacji			
GaussianNB without smoothing	0.0100	0.9466	0.8973
GaussianNB with smoothing	0.4097	0.9466	0.8973
MultinomialNB without smoothing	0.4400	0.8682	0.4908
MultinomialNB with smoothing	0.2962	0.8644	0.4908
K = 2			
GaussianNB without smoothing	0.0359	0.9890	0.9094
GaussianNB with smoothing	0.4790	0.9723	0.9187
MultinomialNB without smoothing	0.3958	0.8581	0.7759
MultinomialNB with smoothing	0.3588	0.8334	0.7853
K = 5			
GaussianNB without smoothing	0.0841	0.9728	0.8995
GaussianNB with smoothing	0.4362	0.9723	0.9000
MultinomialNB without smoothing	0.3720	0.8701	0.7754
MultinomialNB with smoothing	0.3452	0.8357	0.7903
K = 10			
GaussianNB without smoothing	0.0307	0.9781	0.9045
GaussianNB with smoothing	0.4598	0.9781	0.9046
MultinomialNB without smoothing	0.3576	0.8527	0.7807
MultinomialNB with smoothing	0.3515	0.8535	0.7904

Tablica 6: Wpływ kroswalidacji stratyfikowanej na FSC dla zbiorów: *glass*, *wine*, *seeds*

Wnioski

4.1 Obserwacje

Jeżeli chodzi o wygładzanie danych, dla testowanych trzech zbiorów, w większości przypadków przekładało się to na nieznaczną zmianę wyników. Wyjątek stanowi zbiór *glass*, w którym dla modelu `GaussianNB` można zaobserwować znaczną poprawę. Z kolei wygładzanie danych estymowanych przez `MultinominalNB` daje niepożądany efekt pogorszenia miar, mówiących o jakości klasyfikatora.

Rysunki 5, 6, . . . , 22 przedstawiają dane poddane dyskretyzacji. Oczywistą obserwacją jest ta o ich wizualnym uporządkowaniu, co jest potwierdzeniem prawidłowości działania wszystkich badanych metod. W Tabeli 4 zamieszczone zostały dokładne odczyty miary FSC dla każdej z testowanych strategii, każdego zbioru. Można z niej odczytać, że we wszystkich przypadkach dyskretyzacja przynosi oczekiwane rezultaty. Dla zbioru *glass* oraz *seeds* lepiej sprawdza się strategia *uniform* oraz *kmeans*. Natomiast dla zbioru *wine*, dyskretyzacja wydaje się mieć marginalne znaczenie.

W Tabeli 5 oraz 6 umieszczone zostały wyniki przeprowadzenia kroswalidacji na danych. W tym miejscu także możemy zaobserwować pozytywny wpływ stosowania tego narzędzia na wyniki (raczej dla `GaussianNB`). Szczególnie dobrze działa jej stratyfikowana odmiana, która dba o równomierny rozkład klas w zbiorze treningowym i testowym. Warto natomiast zauważyć, że podstawowy

podział danych sprawdza się również w przypadku zbiorów *wine* i *seeds* (tu również dla `GaussianNB`).

4.2 Podsumowanie

Zadanie pozwoliło na zaznajomienie się nie tylko z naiwnym klasyfikatorem Bayesa, ale także z wieloma narzędziami służącymi pracy na danych. Pokazane zostało jak wprowadzać dodatkowe funkcjonalności oraz rozszerzenia, które przekładają się na polepszenie oceny modelu. Wreszcie uświadomiło, że nie istnieje jeden sprawdzony przepis na wszystkie problemy, a tworzenie najlepszych rozwiązań jest związane z odszukaniem właściwego podejścia.