UNIWERSYTET RZESZOWSKI
Kolegium Nauk Przyrodniczych

Paweł Durda

Nr albumu 96449

Informatyka

Analiza metryk i agregacji w zagadnieniach kombinacji klasyfikatora k-NN w przypadku dużej liczby atrybutów w zbiorach danych

Praca magisterska

Praca wykonana pod kierunkiem

dr hab. Urszuli Bentkowskiej, prof. UR

Rzeszów, 2023

RZESZOW UNIVERSITY
College of Natural Sciences

Paweł Durda

Nr albumu 96449

Computer Science

Analysis of metrics and aggregations in k-NN classifier combination problems in the case of a large number of attributes in data sets

Master's Thesis

The thesis written under the supervision of
dr hab. Urszula Bentkowska, prof. UR

Rzeszów, 2023

Pragnę wyrazić moją wdzięczność Pani dr hab. Urszuli Bentkowskiej prof. UR za poświęcony czas i cierpliwość w trakcie realizacji niniejszej pracy magisterskiej. Pani zaangażowanie i życzliwość były dla mnie nieocenionym wsparciem.

Spis treści

1. Wstęp

Celem pracy jest przeprowadzenie analizy znanych metryk i agregacji w celu znalezienia optymalnego sposobu ich zastosowania w algorytmie dedykowanym zbiorom danych o dużej liczbie atrybutów warunkowych. Analiza ta jest przeprowadzona dla algorytmu kombinacji klasyfikatorów *k* najbliższych sąsiadów (k-NN) i dotyczy mikromacierzy, gdzie rozpatrywane metryki oraz agregacje stanowią parametry badanego algorytmu, które mogą istotnie polepszyć jakość klasyfikacji, w konkretnym kontekście algorytmu łączącego wyniki klasyfikacji poszczególnych klasyfikatorów za pomocą agregacji.

Przeglądając literaturę można zauważyć, że najczęściej stosowanymi metrykami odległości w różnych pracach naukowych są metryki Euklidesowa oraz Manhattan, a więc szczególne przykłady metryk z rodziny Minkowskiego. W tej pracy zbadane zostały metryki Minkowskiego z różnymi parametrami oraz metryki Canberra i Bray-Curtis, która jest określana jako znormalizowana metryka Manhattan. Przegląd różnych metryk stosowanych w algorytmach bazujących na algorytmie k-NN oraz ich wpływ na jakość klasyfikacji, można znaleźć w przeglądowej pracy [1]. Dodatkowym parametrem badanym w rozważanym modelu, w związku z zastosowaniem k-NN jest też ilość sąsiadów *k*.

Mikromacierze DNA to narzędzie stosowane w biologii molekularnej do przeprowadzania badań genetycznych. Polegają one na przeniesieniu fragmentów DNA na specjalną matrycę, taką jak np. szkło lub plastikowa karta, za pomocą elektroforezy. Mikromacierze DNA umożliwiają szybkie i wydajne porównywanie genomów różnych organizmów lub różnych stanów fizjologicznych tego samego organizmu. Są one szczególnie przydatne w badaniach nad nowotworami i chorobami genetycznymi. Dane mikromacierzowe charakteryzują się dużą ilością atrybutów warunkowych oraz małą ilością obiektów, co utrudnia klasyfikację takich danych. Ze względu na te ważne zastosowania kluczowe jest przeprowadzanie badań nad polepszaniem jakości metod uczenia maszynowego w kontekście danych mikromacierzowych. Między innymi stosowane są różne metody preprocessingu oraz selekcji najbardziej istotnych atrybutów. W tej pracy zostanie zastosowana metoda selekcji cech RFECV, która jest jedną z zalecanych i częściej stosowanych metod selekcji cech w kontekście mikromacierzy [2].

W badanym modelu, atrybuty wybrane przez RFECV są dzielone na tzw. podtabele, na których uczone są klasyfikatory, więc została zastosowana metoda kombinacji (ang. ensembling) klasyfikatorów. Ilość tych podtabel jest także badana w tej pracy jako jeden z parametrów rozpatrywanego modelu.

Do przeprowadzania eksperymentów na modelu został wykorzystany język programowania Python wraz z biblioteką *scikit-learn* i *scipy.stats,* które posiadają gotowe implementacje różnych klasyfikatorów, najważniejszych metryk oraz agregacji.

Struktura pracy jest następująca:

- Rozdział drugi opisuje podstawy dotyczące klasyfikatorów. W rozdziale tym znajduje się opis algorytmu k-NN oraz metody kombinacji klasyfikatorów.
- Rozdział trzeci jest wprowadzeniem do różnych metryk odległości wykorzystanych w pracy.
- W rozdziale czwartym omówione są różne metody agregacji.
- Rozdział piąty przedstawia metody selekcji cech, które pozwalają na wybór najważniejszych cech dla klasyfikatora.
- W rozdziale szóstym przedstawiony jest schemat badanego algorytmu kombinacji klasyfikatorów oraz sposób jego implementacji i parametry zastosowane w badaniach.
- W rozdziale siódmym opisane są zbiory danych użyte w badaniach.
- W rozdziale ósmym przedstawione są wyniki przeprowadzonych eksperymentów, m.in. porównanie skuteczności badanego modelu do k-NN czy analiza jakości działania modelu względem zastosowanych w nim metryk.
- W ostatnim rozdziale zawarte są wnioski ogólne wynikające z przeprowadzonych badań.

2. Klasyfikacja

2.1. Klasyfikatory

Klasyfikatory są to algorytmy, które służą do rozwiązywania problemu klasyfikacji, czyli przydzielania odpowiedniej kategorii dla danych wejściowych. „Dane wejściowe dla procesu klasyfikacji to tablica decyzyjna, składająca się z obiektów, gdzie każdy obiekt jest reprezentowany przez wektor złożony z atrybutów warunkowych oraz atrybutu decyzyjnego. Zatem system decyzyjny T jest zbiorem obiektów, gdzie każdy obiekt d_n jest reprezentowany przez wektor $\{A_1 = x_1, A_2 = x_2, A_3 = x_3, ..., A_i = x_i, D\}$, gdzie $A = \{A_1, A_2, A_3, ..., A_i\}$ to zbiór atrybutów warunkowych, a $D = \{D_1, D_2, ..., D_j\}$ to atrybut decyzyjny, gdzie $j > 1$. Celem klasyfikacji jest znalezienie funkcji klasyfikacyjnej, która odwzorowuje wartości atrybutów warunkowych obiektów $d_n = \{A_1 = x_1, A_2 = x_2, A_3 = x_3, ..., A_i = x_i\}$ na ich etykiety klas $D = \{D_j\}$. Funkcja klasyfikacyjna może być używana do przewidywania wartości atrybutu decyzyjnego dla nowych obiektów na podstawie znanych wartości atrybutów warunkowych $x_1, x_2, x_3, ..., x_i$ które mogą być zmiennymi ilościowymi lub jakościowymi, natomiast atrybut decyzyjny D musi być zmienną jakościową” ([3], str. 197).

Istnieje wiele różnych algorytmów i metod stosowanych do tworzenia klasyfikatorów, takich jak klasyfikacja na podstawie reguł, drzewa decyzyjne, sieci neuronowe, klasyfikacja bayesowska itp. Każdy z tych algorytmów ma swoje własne zalety i wady, a dobór odpowiedniego algorytmu zależy od charakteru danych wejściowych oraz celu klasyfikacji. W tej pracy został wybrany do analiz algorytm k-NN jako jeden z najbardziej znanych algorytmów wykorzystujących pojęcie metryki odległości.

2.2. Algorytm k-NN

Algorytm k-najbliższych sąsiadów (k-NN) należy do grupy klasyfikatorów leniwych [3], co oznacza, że nie wymaga on wcześniejszego tworzenia modelu. Klasyfikator leniwy trzyma wszystkie dane wejściowe w pamięci i dokonuje przydziału do kategorii dopiero w momencie, gdy jest to konieczne dla nowego obiektu. W przeciwieństwie do klasyfikatorów szybkich, które tworzą model na podstawie danych wejściowych i korzystają z niego do przydziału do kategorii nowych obiektów, klasyfikatory leniwe nie wymagają dodatkowego czasu na proces uczenia, ale ich wydajność może być gorsza, ponieważ muszą przetwarzać wszystkie dane wejściowe za każdym razem, gdy chcemy przydzielić nowy obiekt do kategorii.

Algorytm k-NN działa w następujący sposób. Dla danego obiektu do sklasyfikowania, algorytm oblicza odległość między tym obiektem, a wszystkimi pozostałymi obiektami znajdującymi się w tablicy decyzyjnej. Następnie, algorytm sortuje obiekty według odległości od obiektu do sklasyfikowania i wybiera *k* najbliższych sąsiadów. Algorytm przydziela obiektowi do sklasyfikowania klasę, która jest najczęściej reprezentowana przez jego k najbliższych sąsiadów [3].

W celu polepszenia jakości klasyfikacji za pomocą algorytmu k-NN zalecana jest normalizacja lub standaryzacja danych [3], gdyż bez tego elementu wstępnego przetwarzania danych algorytm k-NN narażony jest na niepożądany wpływ atrybutów o dużych wartościach atrybutów warunkowych (odstających znacznie od pozostałych wartości) na wynik liczonej odległości. W tej pracy została zastosowana normalizacja danych za pomocą klasy *MinMaxScaler* z biblioteki *scikit-learn* [4].

Należy również pamiętać, że czas obliczenia wartości atrybutu decyzyjnego dla nowego obiektu w algorytmie k-NN rośnie wraz z rozmiarem tablicy decyzyjnej, ponieważ dla każdego nowego obiektu musimy porównać jego atrybuty ze wszystkimi atrybutami obiektów znajdujących się w tablicy decyzyjnej, aby znaleźć *k* najbliższych sąsiadów. W przypadku dużych tablic decyzyjnych, obliczenia mogą być bardzo czasochłonne i dlatego algorytm k-NN nie zawsze jest optymalnym wyborem dla dużych zbiorów danych.

2.3. Metoda kombinacji klasyfikatorów

Istnieją różne metody kombinacji klasyfikatorów [3]. Kombinacja klasyfikatorów to metoda polegająca na wykorzystaniu kilku różnych klasyfikatorów do przydzielenia obiektu do kategorii, a następnie wybraniu kategorii na podstawie większości. Możliwe jest również zastosowanie takiej metody, w której każdy z klasyfikatorów ma inną ważność w procesie decyzyjnym (np. niektóre klasyfikatory mogą być bardziej dokładne, ale mniej szybkie, podczas gdy inne są mniej dokładne, ale szybsze).

Kombinacja klasyfikatorów jest często stosowana w celu poprawienia dokładności klasyfikacji, ponieważ w wielu przypadkach różne klasyfikatory mogą charakteryzować się lepszą lub gorszą dokładnością w zależności od charakteru danych wejściowych. Dzięki kombinacji kilku różnych klasyfikatorów możliwe jest wyeliminowanie słabszych punktów poszczególnych klasyfikatorów i uzyskanie wyższej ogólnej jakości klasyfikacji.

Należy jednak pamiętać, że kombinacja klasyfikatorów może prowadzić do wydłużenia czasu obliczeń, ponieważ każdy z klasyfikatorów musi zostać wywołany dla nowego obiektu i jego decyzja musi zostać uwzględniona w procesie decyzyjnym.

W tej pracy zostanie wykorzystana metoda kombinacji klasyfikatorów polegająca na modyfikacji zbioru atrybutów warunkowych danego zbioru danych ([3], str. 316-317). Mianowicie, z oryginalnego zbioru danych treningowych wybierany jest za pomocą danego algorytmu selekcji cech podzbiór atrybutów warunkowych najbardziej istotnych z punktu widzenia jakości klasyfikacji, następnie są one dzielone na *s* podzbiorów treningowych na bazie których konstruowanych jest *s* klasyfikatorów bazowych. Do tej kategorii metod kombinacji klasyfikatorów należy na przykład metoda lasów losowych, w której klasyfikatorami bazowymi są drzewa decyzyjne.

3. Metryki odległości

Przestrzeń metryczna [5] to matematyczny model, w którym każdemu obiektowi przestrzeni *X* przypisana jest odległość *d* od innych obiektów. Ta odległość to jest funkcja postaci:

$$d: X \times X \to [0, +\infty), \tag{1}$$

gdzie, *X* oznacza dowolny niepusty zbiór i funkcja *d* metryka odległości spełnia określone własności dla dowolnych *a, b, c* należących do *X*:

- Identyczność nierozróżnialnych

$$d(a, b) = 0 \leftrightarrow a = b, \tag{2}$$

- Symetria

$$d(a, b) = d(b, a), \tag{3}$$

- Nierówność trójkąta

$$d(a, b) \leq d(a, c) + d(c, b). \tag{4}$$

Warunek nieujemności $d(a, b) \geq 0$ można pominąć przyjmując $d: X \times X \to \mathbb{R}$ zamiast $X \times X \to [0, +\infty)$. Wynika to przedstawionych wyżej aksjomatów, mianowicie:

$$0 = d(a, a) \leq d(a, b) + d(b, a) = 2d(a, b).$$

Przestrzeń metryczna jest często wykorzystywana w uczeniu maszynowym, w tym algorytmie k-NN, gdzie służy do określania podobieństwa obiektów – odległości między nimi. k-NN jest klasyfikatorem opartym na miarach odległości, co oznacza, że im mniejsza odległość między dwoma punktami, tym bardziej są one do siebie podobne. W tej pracy rozważone zostały podstawowe przykłady metryk.

Metryka Minkowskiego (w pracy w skrócie nazywana też jako Minkowski) to uogólniona metryka odległości w przestrzeni wektorowej opisana wzorem [6]:

$$d(x, y) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}}, \tag{5}$$

gdzie *x* i *y* to dowolne punkty w przestrzeni R_n oraz $p \in [1, +\infty)$, $x = (x_1, x_2, \ldots, x_n)$, $y = (y_1, y_2, \ldots, y_n)$.

Warunek trójkąta to jedno z podstawowych założeń, które musi spełniać funkcja odległości, aby była uznana za metrykę. Mówi on, że odległość między dwoma punktami musi być mniejsza lub równa sumie odległości między pierwszym punktem a trzecim punktem oraz odległości między trzecim punktem, a drugim punktem. Do dowodu nierówności trójkąta dla tej metryki wykorzystuje się nierówność Minkowskiego [7]:

$$\left(\sum_{n=1}^{k}|a_n + b_n|^p\right)^{\frac{1}{p}} \leq \left(\sum_{n=1}^{k}|a_n|^p\right)^{\frac{1}{p}} + \left(\sum_{n=1}^{k}|b_n|^p\right)^{\frac{1}{p}}. \quad (6)$$

Parametr p wpływa na kształt i strukturę przestrzeni. Dla *p* mniejszych niż 1, metryka Minkowskiego nie jest odległością, ponieważ nie spełnia warunków przestrzeni metrycznej.

Najczęściej stosowaną metryką w zagadnieniach uczenia maszynowego jest metryka Euklidesowa jako szczególny przypadek metryki Minkowskiego dla *p = 2* opisana wzorem [8]:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}. \quad (7)$$

W metryce Euklidesowej odległość między punktami mierzy się jako długość najkrótszej drogi prowadzącej między nimi, czyli jako odległość liniowa w przestrzeni.

Następną powszechnie stosowaną metryką jest metryka Manhattan, gdzie *p = 1*. Odległość punktów jest sumą bezwzględnych wartości różnicy współrzędnych [9].

$$d(x, y) = \sum_{i=1}^{n}|x_i - y_i|. \quad (8)$$

W przeciwieństwie do metryki Euklidesowej, w metryce Manhattan odległość między dwoma punktami nie jest mierzona jako odległość liniowa, ale jako suma odległości wzdłuż każdej osi współrzędnych. Metryka ta jest bardziej odporna na występowanie wartości odstających niż metryka Euklidesowa [10].

Metryka Czebyszewa mierzy odległość między dwoma punktami jako największą różnicę między ich składowymi. W przypadku, gdy $p \to +\infty$ ze wzoru (5) otrzymujemy metrykę Czebyszewa opisaną wzorem [11]:

$$\mathrm{d}(\mathrm{x}, \mathrm{y}) = \max_{1 \leq i \leq n} |x_i - y_i|. \quad (9)$$

W porównaniu do innych metryk odległości, metryka Czebyszewa uwzględnia tylko największą bezwzględną wartość różnic między odpowiednimi współrzędnymi punktów, co oznacza, że jest ona bardziej wrażliwa na wartości odstające niż inne metryki.

Metryka Canberra nazywana także ważoną wersją metryki Manhattan przedstawia się następującym wzorem [12]:

$$d(x,y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|}. \quad (10)$$

Metryka ta jest bardziej odporna na wartości odstające niż metryka Manhattan, ale z drugiej strony jest mało odporna na wartości bliskie zero [10].

Metryka Bray-Curtis przedstawia się następującym wzorem [13]:

$$d(x,y) = \frac{\sum_{i=1}^{n} |x_i - y_i|}{\sum_{i=1}^{n} |x_i + y_i|}. \quad (11)$$

Metryka ta określana jest jako znormalizowana metryka Manhattan i nazywana jest także w literaturze metryką Sorensena [14].

Do implementacji metryk na potrzeby tej pracy wykorzystana została *sklearn.metrics.DistanceMetric* - klasa w bibliotece *scikit-learn* umożliwiająca wygodne i efektywne obliczanie odległości między próbkami w przestrzeni wielowymiarowej [15]. Klasa ta udostępnia różne miary odległości, w tym metrykę Euklidesową, Manhattan, Canberra, Czebyszewa, Bray-Curtis i wiele innych.

4. Agregacje

Agregacja w badanym klasyfikatorze bazującym na k-NN stosowana jest do łączenia wyników klasyfikacji kilku klasyfikatorów k-NN, które są wyuczone na różnych podzbiorach atrybutów warunkowych. Agregacja $A:[0,1]^n \rightarrow [0,1]$, to n-argumentowa funkcja wielu zmiennych, która jest rosnąca ze względu na każdą ze zmiennych oraz spełnia warunki $A(0,\ldots,0) = 0$, $A(1,\ldots,1) = 1$. Funkcje, których wartości mieszczą się pomiędzy wartościami minimum oraz maximum nazywamy średnimi [16], czyli w przypadku agregacji dla każdego $x_1,\ldots,x_n$ mamy:

$$min(x_1, \ldots, x_n) \leq A(x_1, \ldots, x_n) \leq max(x_1, \ldots, x_n). \tag{12}$$

W badaniach dotyczących tej pracy rozpatrywane były najbardziej znane średnie, mianowicie arytmetyczna, geometryczna, harmoniczna i kwadratowa. Średnia arytmetyczna (w pracy na rysunkach i w tabelach oznaczana też w skrócie Arytmetyczna) wyraża się następującym wzorem:

$$A(x_1, \ldots, x_n) = \sum_{i=1}^{n} w_i x_i. \tag{13}$$

Agregacja (średnia) geometryczna, przedstawiona wzorem (14), jest rzadziej stosowana w badanym kontekście niż agregacja arytmetyczna, ponieważ wynik iloczynu może być bardzo mały dla niewielkich wartości stopni przynależności przydzielanych przez poszczególne klasyfikatory. Może to prowadzić do słabszej dokładności klasyfikacji w porównaniu z innymi metodami agregacji.

$$G(x_1, \ldots, x_n) = \prod_{i=1}^{n} x_i^{w_i}. \tag{14}$$

Agregacja harmoniczna, podobnie jak geometryczna jest rzadziej stosowaną średnią w zagadnieniach klasyfikacji, wyraża się następującym wzorem:

$$H(x_1, \ldots, x_n) = \begin{cases} 0, & \exists\limits_{1 \leq i \leq n} x_i = 0 \\ \frac{1}{\sum_{i=1}^{n} \frac{w_i}{x_i}}, & poza\ tym \end{cases}. \tag{15}$$

Agregacja kwadratowa jest najbardziej znaną z rodziny średnich potęgowych i przedstawiona jest wzorem:

$$Q(x_1, \ldots, x_n) = \sqrt{\sum_{i=1}^{n} w_i {x_i}^2}. \tag{16}$$

W przeprowadzonych badaniach, dla zapewnienia jednakowego wpływu każdego z klasyfikatorów bazowych nie stosowano wag, więc w każdym przypadku agregacji zostało przyjęte $w_i = \frac{1}{n}$ dla każdego *i*.

Warto zauważyć, że rozważane średnie są uporządkowane w następujący sposób [16]:

$$\forall_{x_1, \ldots, x_n \in [0,1]} \; H(x) \leq G(x) \leq A(x) \leq Q(x), \; x = (x_1, \ldots, x_n),$$

co także może mieć wpływ na jakość działania rozważanego algorytmu.

5. Metody selekcji cech

Jednym z wyzwań związanych z uczeniem maszynowym jest obecność dużej liczby cech (zmiennych, atrybutów) w danych wejściowych. W takim przypadku, model uczenia maszynowego może być trudny do trenowania i wymagać dużych zasobów, co może wprowadzać zakłócenia do jego jakości i wydajności. W celu rozwiązania tego problemu stosuje się metody selekcji cech, które pozwalają na zredukowanie liczby cech do tych, które mają największy wpływ na jakość modelu uczenia maszynowego [17].

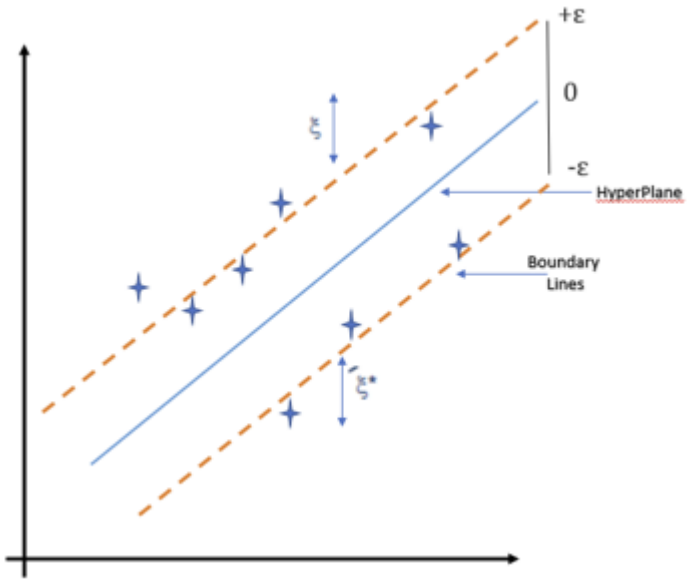
Istnieje wiele różnych metod selekcji cech [2] jak metoda filtrów, metody typu wrapper (ang. wrapper methods), metody wbudowane (ang. embedded methods). Jedna z częściej stosowanych metod selekcji cech RFE lub RFECV, która także została wybrana do zastosowania w tej pracy najczęściej zaliczana jest do metod typu wrapper, czyli bazuje na innym modelu uczenia maszynowego. Można jednak powiedzieć, że technicznie jest to metoda typu wrapper, ale wykorzystuje też metodę filtrów. Modelami uczenia maszynowego wykorzystywanymi w metodzie RFE lub RFECV są często SVM i SVR.

Support Vector Machine (SVM) jest jedną z metod uczenia maszynowego, która jest szeroko stosowana w problemach klasyfikacji. Celem SVM jest wyznaczenie optymalnej hiperpłaszczyzny, która jak najlepiej oddziela dane treningowe na podstawie ich etykiet klasy. Hiperpłaszczyzna ta jest wyznaczana tak, aby zachować największą odległość między poszczególnymi przykładami z różnych klas [18].

Support Vector Regression (SVR) jest jedną z metod uczenia maszynowego, która polega na wyznaczeniu optymalnej hiperpłaszczyzny w danych, która jak najlepiej oddziela dane treningowe na podstawie ich wartości. SVR jest modyfikacją popularnej metody SVM, która jest zwykle stosowana w problemach klasyfikacji. W przeciwieństwie do klasyfikacji, SVR jest używany w problemach regresji, co oznacza, że jego celem jest przewidywanie wartości ciągłych na podstawie danych treningowych [19].

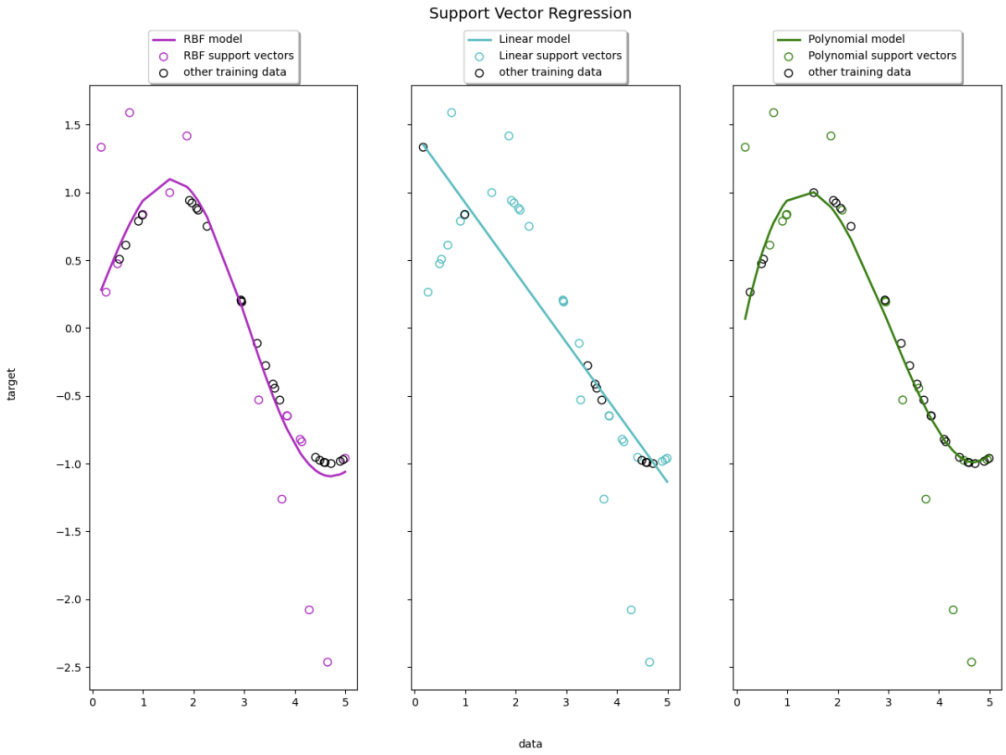
Estymator SVR jest parametrem, który określa sposób, w jaki model SVR będzie trenowany i dokonywał predykcji. Estymator SVR może być używany z różnymi jądrami (kernelami) do modelowania danych i ma kilka hiperparametrów, takich jak parametr *C*, które mogą wpływać na jakość modelu. Wartość parametru *C* jest ważna dla SVR, ponieważ odpowiada za kompromis między minimalizacją odchyleń a liczbą punktów wewnątrz

hiperpłaszczyzny. Im większa wartość *C*, tym bardziej skupia się na minimalizacji odchyleń, co może prowadzić do bardziej skomplikowanego modelu [20].



Rysunek 1 Wizualizacja działania algorytmu SVR źródło https://www.educba.com/support-vector-regression/ 06.02.2023

Parametr Kernel pozwala na wybór hiperpłaszczyzny dzielącej dane. Sklearn library oferuje różne opcje kerneli, takie jak *linear*, *polynomial*, *RBF*.



Rysunek 2 Wizualizacja różnicy w wyborze kerneli RBF, Linear, Polymial źródło https://scikit-learn.org/stable/auto_examples/svm/plot_svm_regression.html 06.02.2023

SVR jest często stosowany w przypadku danych o wysokiej liczbie wymiarów lub gdy istnieje duża liczba szumów lub elementów odstających w danych. Może być również używany do rozwiązywania problemów, w których istnieje duża liczba cech, ale mało przykładów danych treningowych.

RFE (ang. Recursive Feature Elimination) to metoda selekcji cech, która jest często stosowana w uczeniu maszynowym. Celem tej metody jest identyfikacja i usunięcie cech, które są nieistotne dla danego problemu, aby uzyskać lepsze modele [21]. RFE działa poprzez iteracyjne usuwanie cech i ocenianie wpływu na jakość modelu. Proces rozpoczyna się od uwzględnienia wszystkich cech, a następnie usuwanie najmniej istotnej cechy w każdej iteracji. Jakość modelu jest oceniana za pomocą metryk takich jak dokładność, *AUC* itp. Proces ten jest powtarzany, aż do osiągnięcia zadanej liczby cech lub osiągnięcia satysfakcjonującej jakości modelu. RFE jest często stosowane w połączeniu z innymi metodami uczenia maszynowego, takimi jak regresja liniowa, drzewa decyzyjne i SVM.

Metoda Selekcji cech RFECV (Recursive Feature Elimination with Cross-Validation) jest to narzędzie do automatycznej selekcji cech (zmiennych, atrybutów) w modelu uczenia maszynowego. Polega na rekurencyjnym usuwaniu cech z modelu i ocenianiu jego

skuteczności za pomocą walidacji krzyżowej. Cechy są usuwane od tej o najmniejszym wpływie na skuteczność modelu, a proces jest kontynuowany aż do osiągnięcia określonego progu skuteczności lub do momentu, gdy wszystkie cechy zostały wypróbowane. W ten sposób można znaleźć optymalny zestaw cech dla danego modelu. RFECV jest szczególnie przydatne w przypadku dużych zbiorów danych, gdzie istnieje duża liczba cech do wyboru, ponieważ automatycznie wybiera najlepsze cechy i pozwala uniknąć przetrenowania (overfitting) modelu. Jest to również użyteczne, gdy nie jest jasne, które cechy są najważniejsze dla danego problemu lub gdy chcemy zmniejszyć złożoność modelu i poprawić jego interpretowalność. Podsumowując wszystkie informacje na temat selekcji cech, w tej pracy został wybrany do zastosowania model RFECV z estymatorem SVR. Opis parametrów RFECV z biblioteki *scikit-learn* przedstawia Tabela 1.

Tabela 1 Parametry dla sklearn.feature_selection.RFECV [22]

Parametr	Opis
estimator	estymator, który zostanie wykorzystany do selekcji cech.
step	liczba cech, o jaką ma być zmniejszana liczba cech w każdej iteracji selekcji. Domyślnie wartość to 1.
min_features_to_select	minimalna liczba cech, która ma być wybrana
cv	specyfikacja strategii walidacji krzyżowej, domyślnie używana jest 5-krotna walidacja krzyżowa.
scoring	określa metrykę, która będzie używana do oceny jakości modelu.
verbose	określa, jak wiele komunikatów ma być wyświetlanych podczas wykonywania walidacji krzyżowej.
n_jobs	liczba rdzeni procesora, które mają zostać użyte do obliczeń.
importance_getter	parametr pozwala na wybór sposobu obliczania ważności cech w procesie eliminacji wstecznej, domyślnie wykorzystywana jest metoda coef.

6. Implementacja algorytmu

6.1. Schemat algorytmu kombinacji klasyfikatorów

Rozważany klasyfikator jest binarny. W pracy wykorzystano algorytm RFECV, który pozwala na automatyczne wybranie najważniejszych genów dla danych mikromacierzowych (pominięcie cech z małą ilością informacji), co pozwala na zwiększenie skuteczności prognozowania oraz zwiększenie efektywności uczenia maszynowego, poprzez redukcję liczby cech. RFECV jest szczególnie przydatny w przypadku, gdy liczba cech jest duża w stosunku do liczby próbek. Dodatkowo został zastosowany podział zbioru cech na tabele, których ilość określa parametr *s*. Dla każdej tabeli został utworzony klasyfikator k-NN, który w oparciu o wybrane metryki oraz agregacje zostanie użyty jako podstawa do utworzenia kombinacji klasyfikatorów badanego modelu. Wynik klasyfikacji w tym przypadku będzie agregacją (średnią) współczynników przynależności do klasy głównej uzyskanych dla poszczególnych klasyfikatorów. Działanie algorytmu przedstawione zostało schematycznie na *Rysunku 3*.

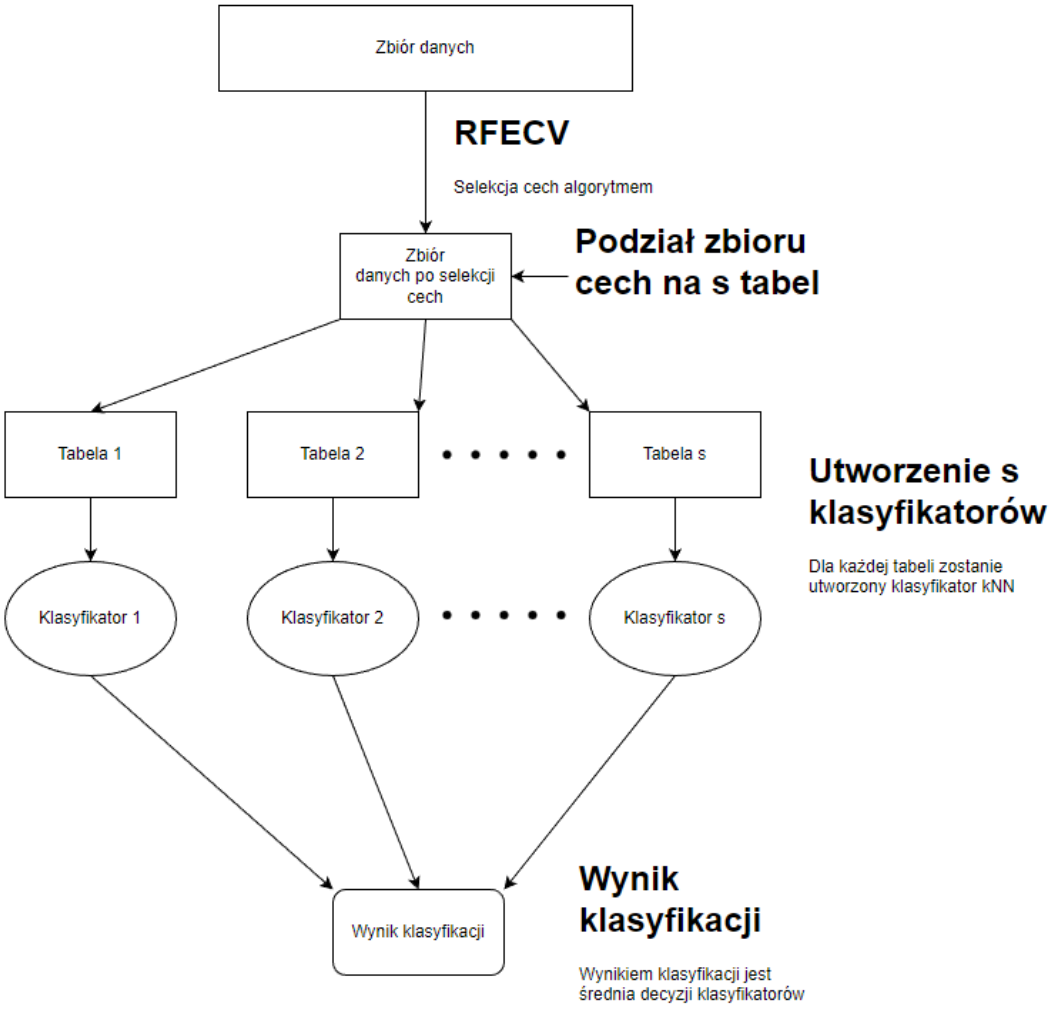
Algorytm ma następujące parametry: metryka, agregacja, ilość podtabel, ilość sąsiadów *k* w algorytmie k-NN oraz dodatkowo różne wartości parametru *p* w przypadku metryki Minkowskiego. W szczególności zastosowano:

- Metryki
  - Euclidean
  - Manhattan
  - Czebyszew
  - Canberra
  - Bray-Curtis
  - Minkowski
- Agregacje
  - arytmetyczna
  - geometryczna
  - harmoniczna
  - kwadratowa
- parametr *s* (ilość podtabel, czyli klasyfikatorów bazowych)
  - 2
  - 5

- 10
- 20

- Ilość sąsiadów (*n_neighbors*)
  - 3
  - 5
  - 7

- parametr dla metryki Minkowskiego (*p*)
  - 1,5
  - 3
  - 5
  - 10
  - 20



Rysunek 3 Diagram przedstawiający ogólny schemat algorytmu

6.2. Szczegóły eksperymentów

W pracy zastosowano język Python z bibliotekami *scikit-learn* i *numpy* do implementacji algorytmów. Model jest zapisany w pliku *model.py*, który korzysta z agregacji z pliku *agregacje.py*. Implementacja klasy *Model*, która przyjmuje parametry opisane w tabeli poniżej, jest zawarta w pliku *main.py*.

Tabela 2 Opis parametrów dla klasy implementującej Model

Parametr	Opis
n_neighbors	liczba sąsiadów (domyślnie 3) używana do klasyfikacji k-NN
metric	metryka używana do obliczania odległości między punktami danych (domyślnie 'minkowski')
p	parametr określający rodzaj metryki 'minkowski'
s	liczba określająca ilość podtabel
t	parametr określający próg dla przypisania obiektów do głównej klasy. Jego wartość musi być z przedziału (0;1)
aggregation	metoda agregacji klasyfikatorów k-NN
RFECVestimator	parametr określający wybrany estymator
RFECVstep	parametr step określa co ile cech ma być usuniętych w procesie eliminacji rekurencyjnej
RFECV_min_n_features	minimalna liczba cech, która ma być wykorzystana przy wyborze cech za pomocą RFECV (domyślnie 200)

Dla każdego zbioru określana była minimalna liczba cech w wartości 5% ilości cech, jako parametr dla *RFECV_min_n_features*.

Do podziału zbioru danych na część treningową i testową, użyto krosswalidacji stratyfikowanej. W tej metodzie zbiór danych jest dzielony na *k* równych części (foldów), ale każdy z tych foldów zawiera proporcjonalną reprezentację różnych kategorii zmiennych [3].

```
skf = StratifiedKFold(n_splits=10)
skf.get_n_splits(X, y)

def create_models(estimator=None):···

def pred_models(models_SVR):
    samplesSVR = {}
    for train_index, test_index in skf.split(X, y):
        X_train, X_test = X[train_index], X[test_index]
        y_train, y_test = y[train_index], y[test_index]
```

Rysunek 4 Krosswalidacja zbioru danych na 10 części

W metodzie *get_sample()* zostanie zwrócony zbiór cech podzielony na podtabele. W wyniku działania algorytmu RFECV uzyskano listę wybranych cech, która została przetasowana i podzielona na *s* części. Następnie cechy te były wstawiane w kolejności do podtabel.

```
def get_sample(X, y, n_split, selector, iter, name_estimator):
    # ranking = dict(enumerate(selector.ranking_.flatten(), 0))
    # ranking_best = {key: val for key, val in ranking.items() if val == 1}
    samples = {}
    features = dict(enumerate(selector.get_support().flatten(), 0))
    # print(features)
    features_true = {key: val for key, val in features.items() if val == True}
    features_keys = list(features_true.keys())
    random.shuffle(features_keys)
    # print('features_keys', features_keys)
    # if len(features_keys) > percentage_of_set:
    #     features_keys = random.sample(features_keys, percentage_of_set)
    # print('features_keys sample', features_keys, len(features_keys))
    features_keys = list(split_list(features_keys, n_split))
    # print(features_keys)
    for i in range(len(features_keys)):
        samples[i] = X[:, features_keys[i]]
        samples['features_true_' + str(i)] = features_keys[i]
    print('STATS: StratifiedKFold Iter ', str(iter), 'Estimator ', name_estimator, 'S: ', n_split,
          'count features true ', len(features_true), 's ', len(features_keys))
    return samples

def split_list(seq, size):
    return (seq[i::size] for i in range(size))
```

Rysunek 5 Metoda tworząca podział na podtabele

```
samples: {0: array([[0.22586308, ...7825357]]), 'features_true_0': [2222, 5771, 3839, 5832, 4513, 295, 1248, 262, 3550, ...], 1: a…
  special variables
  function variables
  0: array([[0.22586308, 0.34734735, 0.45323741, ..., 0.28042328, 0.        ,
  'features_true_0': [2222, 5771, 3839, 5832, 4513, 295, 1248, 262, 3550, 4190, 4210, 5580, 1480, 3251, ...]
  1: array([[0.05247525, 0.43908629, 0.46393252, ..., 0.45230525, 0.51877133,
  'features_true_1': [4938, 3210, 4152, 1106, 6280, 6400, 5715, 1974, 2617, 3925, 1996, 1614, 1598, 2597, ...]
  len(): 4
```

Rysunek 6 Przykładowy wgląd do słownika samples, dla podziału na dwie podtabele.

Proces uczenia się klasyfikatorów został podzielony na dwie metody przyjmujące różne parametry. W przypadku metody *fit()* klasyfikator jest uczony na zbiorze cech podzielonych na wcześniej zadeklarowaną ilość podtabel, w przeciwieństwie do metody *fit_single_knn()*, gdzie system decyzyjny zachowuje pierwotny rozmiar.

```
def fit_single_knn(self, X, y, sample_dict):
    selected_features = []
    for key, val in sample_dict.items():
        if type(key) is not int:
            selected_features.extend(val)
```

Rysunek 7 Wywołanie metody fit_single_knn()

Ostatni etap implementacji polega na ocenie jakości klasyfikatora. Metoda *score()* służy do oceny jakości klasyfikacji. Na początku wykorzystywana jest metoda *predict(),* która przewiduje klasy dla nowych danych wejściowych. Pierwszy krok metody to zastosowanie metody *predict_proba(),* która przewiduje prawdopodobieństwo przynależności do każdej z klas dla nowych danych wejściowych. Następnie wykorzystywana jest jedna z metod agregacji (*amean*, *qmean*, *gmean* lub *hmean*) do połączenia wyników predykcji dla każdego klasyfikatora w celu otrzymania pojedynczego wyniku. Wynik ten jest porównywany z progiem *t* i przypisywana jest klasa. Wyniki te są używane do aktualizacji macierzy pomyłek, z której można obliczyć wiele miar jakości klasyfikacji, takich jak dokładność (accuracy). Na potrzeby opisu jakości badanego modelu przyjęto miarę jakości *AUC*, przy czym na podstawie wartości zwracanych w każdym *foldzie* wyznaczano średnią wartość *AUC* oraz odchylenie standardowe. Metody *predict_for_single_knn()* oraz *score_for_single_knn()* są wywoływane analogicznie, ale tylko dla pojedynczego modelu k-NN.

Input:
System decyzyjny T
Parametry modelu:
- metric
- aggregation
- n_neighbors
- p
- RFECVstep
- RFECV_min_n_features
- RFECVestimator
- t

Output:
Wyniki stratyfikowanej kroswalidacji modelu

Begin:
Wczytanie i przygotowanie danych
Podział stratyfikowany na dziesięć części
For i:=1 **to** 10 **do**:
Trenowanie klasyfikatora na pozostałych dziewięciu częściach zbioru
Ocena klasyfikatora na i-tej części zbioru
Zapisanie wyniku oceny klasyfikatora
End
Zapis wyników do pliku
End

Rysunek 8 Pseudokod opisujący główne kroki algorytmu

7. Zbiory danych

Mikromacierze to zbiory danych, które zawierają informacje o poziomie ekspresji genów w różnych warunkach lub różnych grupach komórek. Cechują się one dużą liczbą genów (nawet kilka tysięcy) i małą liczbą próbek (nawet kilkanaście). Eksperymenty dotyczące analizy mikromacierzy zostały przeprowadzone z wykorzystaniem pięciu różnych zbiorów danych. Celem tych eksperymentów na różnych zbiorach danych było upewnienie się, że otrzymane wyniki nie są pojedynczymi przypadkami zależnymi od zbioru danych. Oryginalne dane zostały pobrane z *ELVIRA Biomedical Data Set Repository* [23] oraz zostały przetworzone do formatu *csv*.

Pierwszym użytym zbiorem danych jest Colon [24]. Zawiera on informacje dotyczące poziomów ekspresji genów w próbkach pobranych od pacjentów z rakiem jelita grubego. Zbiór składa się z 62 próbek, z których 40 pochodzi z guzów nowotworowych, a pozostałe 22 pochodzą ze zdrowych części jelita grubego. Spośród około 6500 genów, 2000 zostało wybranych na podstawie jakości pomiarów poziomów ekspresji.

Drugim zbiorem jest Lymphoma [25]. Zawiera on informacje dotyczące osób cierpiących na chłoniaka rozlanego. Zbiór składa się z 47 próbek, z czego 24 należy do klasy *germinal*, a pozostałe 23 do klasy *activated*. Każda z próbek jest opisana przez pomiary poziomów ekspresji 4026 genów.

Kolejnym zbiór Leukemia zawiera informacje dotyczące dwóch typów białaczki: ostrej białaczki limfocytowej (*ALL*) oraz ostrej białaczki mielocytowej (*AML*) [26]. Dane dotyczące ekspresji genów zostały pobrane ze szpiku kostnego osób chorych. Zbiór składa się z 72 próbek, z czego 25 z nich jest klasy *AML*, a pozostałe 47 jest klasy *ALL*. Każda próbka jest opisana przez 7129 cech.

Następnym z testowanych zbiorów jest Ovarian [27]. Zbiór został stworzony w celu identyfikacji wzorców proteomicznych w surowicy, co pozwalałoby na odróżnianie zdrowych osób od chorych na raka jajnika. Jest to szczególnie istotne dla kobiet z wysokim ryzykiem zachorowania, takich jak te, które mają w rodzinie osoby chore na raka jajnika. Zbiór składa się z 91 próbek kontrolnych uważanych za prawidłowe i 162 próbek chorych na raka jajnika. Każda z próbek jest opisana przez 15154 atrybutów.

Zbiór danych Prostate zawiera informacje na temat raka prostaty [28]. Zawiera on dwa zbiory: treningowy z 52 próbkami pobranymi od chorych i zdrowych osób oraz testowy z 25 próbkami chorych i 9 zdrowymi. Wszystkie próbki mają taką samą liczbę 12600 atrybutów.

Podsumowanie informacji dotyczących zbiorów danych wykorzystanych w eksperymencie znajduje się w Tabeli 3.

Tabela 3 Opis zbioru danych

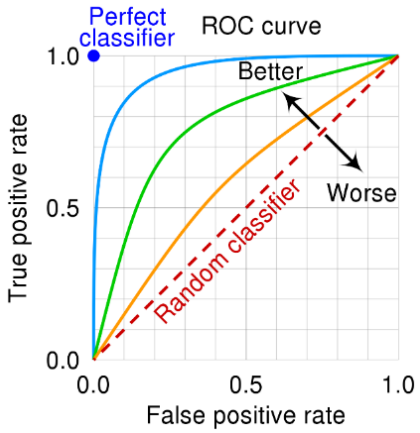
Nazwa zbioru	Liczba obiektów	Liczba atrybutów	Klasy decyzyjne	Czego dotyczy
Colon	62	2000	2	Nowotwór okrężnicy
Leukemia	72	7129	2	Białaczka
Lymphoma	47	4026	2	Chłoniak
Ovarian	253	15154	2	Nowotwór jajnika
Prostate	136	12600	2	Nowotwór Prostaty

8. Analiza wyników

Wyniki klasyfikacji zostały ocenione względem kilku miar jakości, w tym *AUC*, odchylenie standardowe *AUC*, *accuracy* (dokładność) - stosunek liczby poprawnie sklasyfikowanych obiektów do całkowitej liczby obiektów, odchylenie standardowe dla *accuracy* oraz *FP (False Positive), FN (False Negative), TP (True Positive)* i *TN (True Negative)* to wartości liczbowe reprezentujące ilość fałszywie pozytywnych, fałszywie negatywnych, prawdziwie pozytywnych i prawdziwie negatywnych sklasyfikowań w modelu (por. [3]).

Ostatecznie, do oceny i analizy modelu zdecydowano się posługiwać miarą jakości *AUC*. Area Under the Curve (AUC) to miara jakości w uczeniu maszynowym, która ocenia jakość modelu klasyfikacji [3]. Warto zaznaczyć, że proces selekcji cech za pomocą algorytmu RFECV był przeprowadzany w każdym z foldów z osobna, przy czym różna liczba cech (nie mniejsza niż zadeklarowana minimalna liczba cech) mogła zostać wybrana dla poszczególnych foldów. Pozostałe kroki algorytmu również były prowadzone w każdym z foldów z osobna. Dlatego wyznaczane było średnie AUC i poddawane analizie.

AUC jest to powierzchnia pod krzywą *ROC* (ang. Receiver Operating Characteristic), która jest wykresem prezentującym zależność między *True Positive Rate (TPR),* a *False Positive Rate (FPR)* w modelu klasyfikacji binarnej przy różnych wartościach progu w wykonywanym eksperymencie.



Rysunek 9 Wizualizacja dla przestrzeni ROC dla „lepszego" i „gorszego" klasyfikatora źródło https://en.wikipedia.org/wiki/Receiver_operating_characteristic 06.02.2023

Wartość *AUC* zawiera się w zakresie od 0 do 1, gdzie wartość 1 oznacza idealny model klasyfikacji, a wartość 0.5 oznacza model losowy. Wartość *AUC* jest często wykorzystywana do porównywania różnych modeli klasyfikacji i wyboru najlepszego z nich.

W pracy nie zostaną podane szczegółowe wyniki dotyczące jakości klasyfikacji, z tego względu iż arkusz dla każdego zbioru danych liczy 480 wierszy. Jednak cząstkowe wyniki zostały zebrane w tabelach i opisane w dalszej części pracy.

8.1. Porównanie skuteczności badanego modelu do k-NN

Ze względu na to, że rozważany w pracy model stanowi kombinację klasyfikatorów, wykonano testy statystyczne, które miały na celu porównanie badanego modelu do pojedynczego modelu k-NN. Porównane zostały wyniki pięciu zestawów danych, tj. *Colon*, *Prostate*, *Leukemia*, *Ovarian* i *Lymphoma* bez rozróżniania na poszczególne parametry badanego modelu kombinacji klasyfikatorów (takie jak na przykład ilość podtabel, metryka czy agregacja). Wyniki przedstawiono w Tabeli 4. Widać, że badany model biorąc pod uwagę wszystkie rozważane parametry niestety osiąga gorszą lub taką samą jakość jak pojedynczy model. Aby ocenić istotność statystyczną różnic między średnimi wartościami dwóch niezależnych próbek (badanego modelu i odpowiadającego mu modelu k-NN), do obliczeń wykorzystano średnie *AUC* oraz odchylenie standardowe porównywanych modeli zawarte w Tabeli 4.

Wyniki testów statystycznych wskazują, że badany model kombinacji klasyfikatorów nie osiąga lepszych wyników lub osiąga wyniki porównywalne z pojedynczym modelem k-NN dla wszystkich rozważanych zestawów danych (Colon, Prostate, Leukemia, Ovarian i Lymphoma), bez względu na rozważane parametry badanego modelu kombinacji klasyfikatorów (takie jak na przykład ilość podtabel, metryka czy agregacja). Analizując dane przedstawione w Tabeli 4, można zauważyć, że średnie *AUC* dla badanej kombinacji klasyfikatorów są zawsze niższe od *AUC* dla pojedynczego modelu k-NN. Odchylenie standardowe dla badanej kombinacji klasyfikatorów jest zwykle większe niż dla pojedynczego modelu k-NN, co może sugerować większą niestabilność wyników dla badanego modelu.

Tabela 4 Wyniki mediany oraz Średnie AUC dla zbiorów danych

Zbiór danych	Średnia AUC kombinacja klasyfikatorów	Średnia AUC pojedynczy k-NN	Odchylenie standardowe kombinacja klasyfikatorów	Odchylenie standardowe pojedynczy k-NN
Colon	0,89152	0,93799	0,11236	0,03184
Prostate	0,92611	0,97478	0,08792	0,02411
Leukemia	0,99166	0,99433	0,02096	0,00803
Ovarian	0,9926	0,99991	0,01539	0,00019
Lymphoma	0,95904	1	0,07448	0

Test Manna-Whitneya to jedno z najczęściej stosowanych narzędzi w analizie statystycznej, szczególnie wtedy, gdy nie można spełnić założeń dotyczących normalności rozkładu danych. Ten test nieparametryczny pozwala na porównanie dwóch niezależnych prób, a wyniki testu są wyrażane w postaci statystyki *U* i wartości *p* [29], [30].

Wynik testu dla poszczególnych zestawów danych przedstawiają się następująco: *Ovarian: p = 0.00082521, Leukemia: p = 0.50686340, Colon: p = 0.50105408, Prostate: p = 0.00006098, Lymphoma: p = 0.00010177.* W tym przypadku, wartości *p* dla zestawów *Ovarian*, *Prostate* i L*ymphoma* sugerują istotne różnice pomiędzy średnimi wartościami *AUC* dla badanych modeli, podczas gdy wartości *p* dla zestawów Colon i Leukemia nie są na tyle niskie, aby odrzucić hipotezę o równości średnich *AUC* badanych modeli. Tabela 4 zawiera wartości odchylenia standardowego i średnich *AUC* dla badanego modelu (kombinacja klasyfikatorów) oraz pojedynczego k-NN. W Tabeli 5 zebrane zostały wartości *p* dla wspomnianego wyżej testu.

Różnice w wynikach między pojedynczym k-NN, a kombinacją klasyfikatorów na niekorzyść modelu kombinacji klasyfikatorów, wynikają z dysproporcji wyników uzyskanych z różnych kombinacji parametrów, takich jak metryki, agregacja lub ilość podtabel itd. Wskazuje na to odchylenie standardowe z Tabeli 4 proponowanego modelu, ponieważ było zwykle wyższe niż pojedynczego modelu k-NN, co wskazuje na większą niestabilność wyników. Dlatego średnia liczona dla kombinacji klasyfikatorów wypada gorzej.

Tabela 5 Wyniki statystyczne dla zbiorów danych

Zbiór danych	Wartość p dla testu Manna-Whitneya
Colon	0,50105408
Prostate	0,00006098
Leukemia	0,50686340
Ovarian	0,00082521
Lymphoma	0,00010177

W kolejnych podrozdziałach zostanie przeprowadzona bardziej szczegółowa analiza badanego modelu, w szczególności porównanie wyników modelu dla metryk względem ilości podtabel i sąsiadów, a także względem rozpatrywanych agregacji. Przeprowadzone analizy pokażą, które wartości ze wspomnianych parametrów znacznie zaniżają jakość badanego modelu.

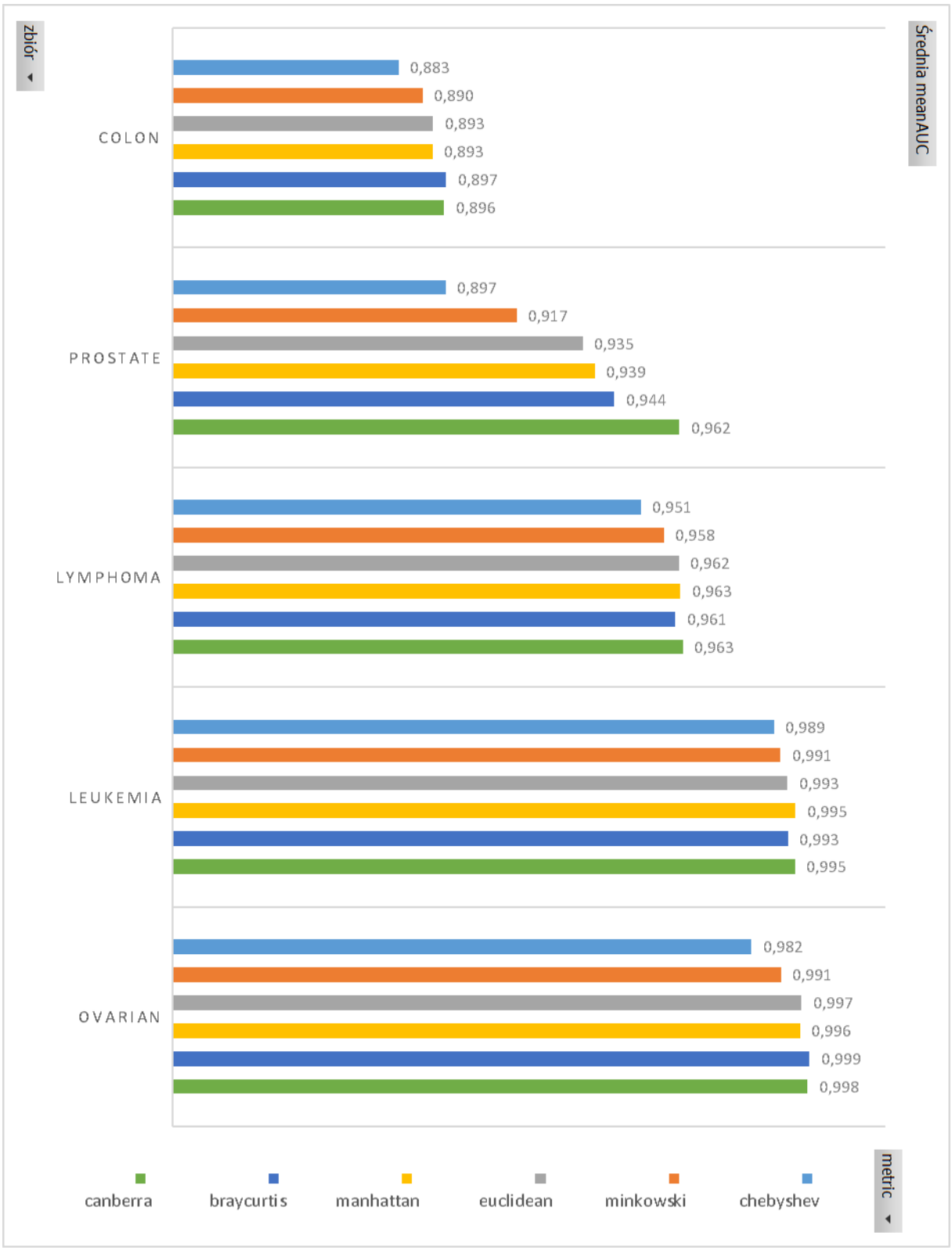
8.2. Porównanie modelu dla metryk względem ilości podtabel i sąsiadów

W tym rozdziale na początku zostaną porównane wyniki *AUC* modelu dla poszczególnych metryk i zbiorów danych. Na Rysunku 10 oraz w Tabeli 6 zebrane są wyniki *AUC* modelu dla poszczególnych zbiorów danych i metryk. Można zauważyć, że nie biorąc pod uwagę dodatkowych parametrów modelu (ilość podtabel i sąsiadów) metryki na odpowiednich zbiorach danych dają dość podobne wyniki *AUC*. Najniższa jakość modelu została uzyskana dla zbioru danych Colon.

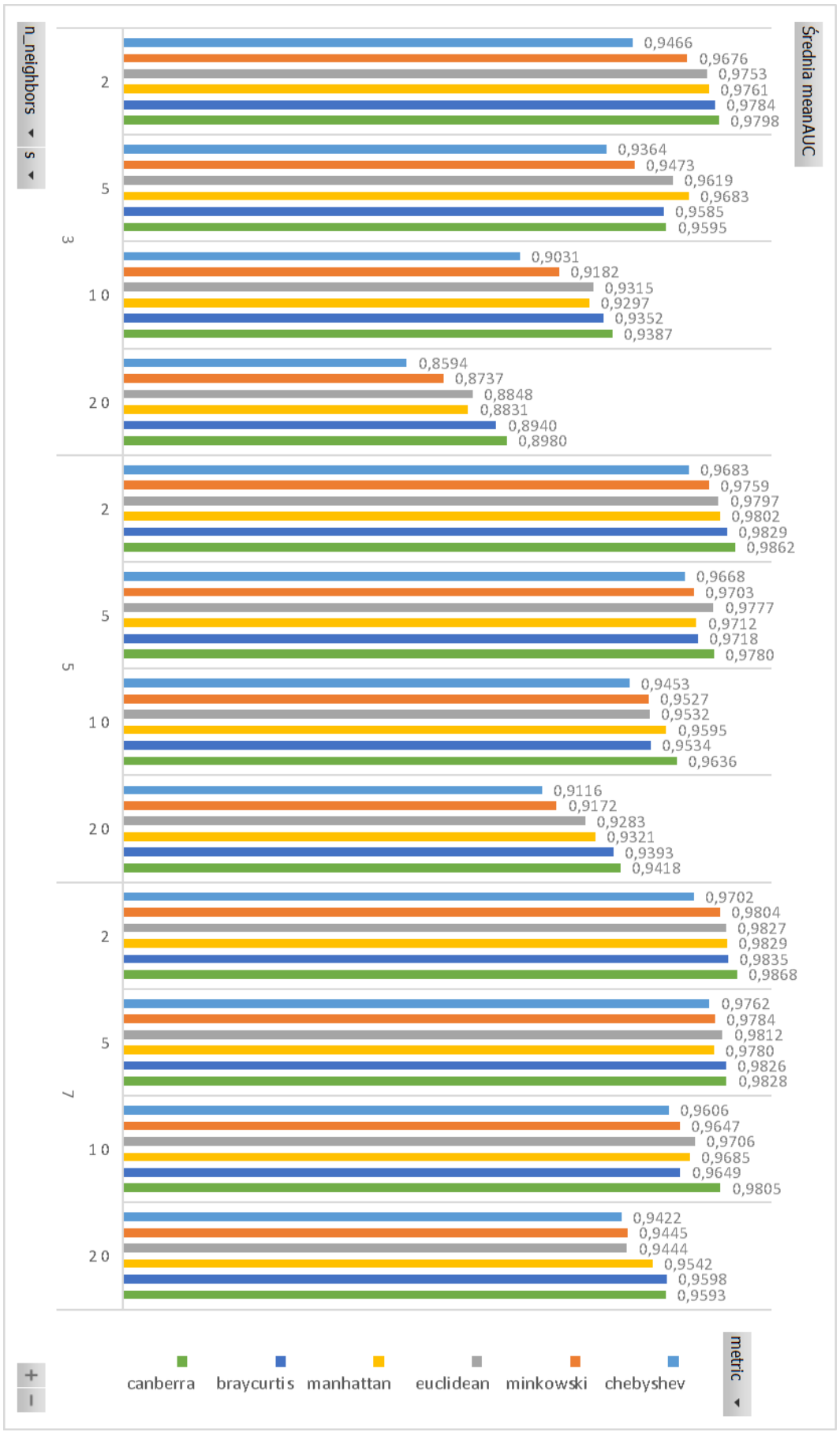
Tabela 6 Średnie AUC modelu dla zbiorów danych i metryk

Średnie AUC	Chebyshev	Minkowski	Euclidean	Manhattan	Braycurtis	Canberra
Colon	0,883	0,890	0,893	0,893	0,897	0,896
Prostate	0,897	0,917	0,935	0,939	0,944	0,962
Lymphoma	0,951	0,958	0,962	0,963	0,961	0,963
Leukemia	0,989	0,991	0,993	0,995	0,993	0,995
Ovarian	0,982	0,991	0,997	0,996	0,999	0,998

Można jednak zauważyć, że niektóre metryki jako parametr badanego modelu, dodatkowo z uwzględnieniem parametrów jakimi są ilość podtabel i sąsiadów, jednak bez rozróżniania na poszczególne zbiory danych, skutkowały uzyskaniem lepszych średnich wyników *AUC*. Różnice te można zaobserwować na Rysunku 11 i w Tabeli 7.



Rysunek 10 Porównanie średnich AUC dla zbiorów danych i metryk



Rysunek 11 Porównaniem AUC dla metryk, ilości podtabel i ilości sąsiadów

Tabela 7 Średnie AUC modelu dla metryk, ilości podtabel s i ilości sąsiadów k

Średnie AUC	Chebyshev	Minkowski	Euclidean	Manhattan	Bray-Curtis	Canberra
k=3						
2	0,9466	0,9676	0,9753	0,9761	0,9784	**0,9798**
5	0,9364	0,9473	0,9619	**0,9683**	0,9585	0,9595
10	0,9031	0,9182	0,9315	0,9297	0,9352	**0,9387**
20	0,8594	0,8737	0,8848	0,8831	0,8940	**0,8980**
k=5						
2	0,9683	0,9759	0,9797	0,9802	0,9829	**0,9862**
5	0,9668	0,9703	0,9777	0,9712	0,9718	**0,9780**
10	0,9453	0,9527	0,9532	0,9595	0,9534	**0,9636**
20	0,9116	0,9172	0,9283	0,9321	0,9393	**0,9418**
k=7						
2	0,9702	0,9804	0,9827	0,9829	0,9835	**0,9868**
5	0,9762	0,9784	0,9812	0,9780	0,9826	**0,9828**
10	0,9606	0,9647	0,9706	0,9685	0,9649	**0,9805**
20	0,9422	0,9445	0,9444	0,9542	**0,9598**	0,9593

W Tabeli 7 pogrubieniem zostały wyróżnione największe wartości w danym wierszu tej tabeli. Na podstawie Tabeli 7 widać, że metryka Canberra najczęściej jest tą, której zastosowanie skutkuje najwyższą wartością *AUC*. Metryka Canberra dała też najlepsze wyniki *AUC*, biorąc pod uwagę średnie wartości dla wszystkich przypadków (0,96291).

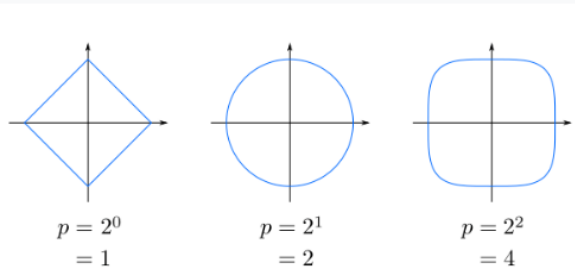
Dodatkowo, analizując wyniki na Rysunku 11, widać, że niezależnie od ilości sąsiadów *k* im więcej podtabel *s*, tym wynik *AUC* jest niższy. Najstabilniejsze wyniki uzyskano dla sąsiedztwa o wartości 7, to znaczy spadek *AUC* dla większych wartości *s* był przy tej ilości sąsiadów mniejszy.

Metryki, które zastosowane jako parametr modelu skutkowały uzyskaniem najlepszych wartości *AUC*, to metryka Minkowski z parametrem $p = 1.5$, metryka Euklidesowa, metryka Manhattan, metryka Canberra, metryka Bray-Curtis. W celu potwierdzenia tych obserwacji i ewentualnego wyłonienia jakiejś najlepszej metryki został przeprowadzony test statystyczny. Metryki Chebyszewa oraz metryk Minkowskiego z parametrem $p > 1.5$ nie wzięto pod uwagę w przeprowadzonym teście, aby nie zaburzać jego wyniku. Badany model przy zastosowaniu metryki Chebyszewa oraz metryk Minkowskiego z parametrem $p > 1.5$ uzyskał wyraźnie najgorsze wyniki *AUC*.

Testu Kruskala-Wallisa (H-test) [31] użyto do porównania istotności różnic średnich *AUC* pomiędzy poszczególnymi wersjami modelu z zastosowanymi metrykami (metryka Minkowski z parametrem *p = 1.5*, metryka Euklidesowa, metryka Manhattan, metryka Canberra, metryka Bray-Curtis) oraz z uwzględnieniem różnej ilości podtabel, sąsiadów i agregacji. Kruskal-Wallis H-test jest testem nieparametrycznym, który porównuje kilka grup na podstawie ich średnich rang. Test ten jest używany do wykrywania istotnych różnic między kilkoma grupami, gdy nie można założyć normalności rozkładu danych. Wartość statystyczna H jest obliczana na podstawie rang danych, a wartość *p* jest obliczana na podstawie wartości H i liczby grup, aby stwierdzić, czy istnieją istotne różnice między grupami, a dokładnie, czy jakaś para analizowanych danych wyróżnia się spośród pozostałych.

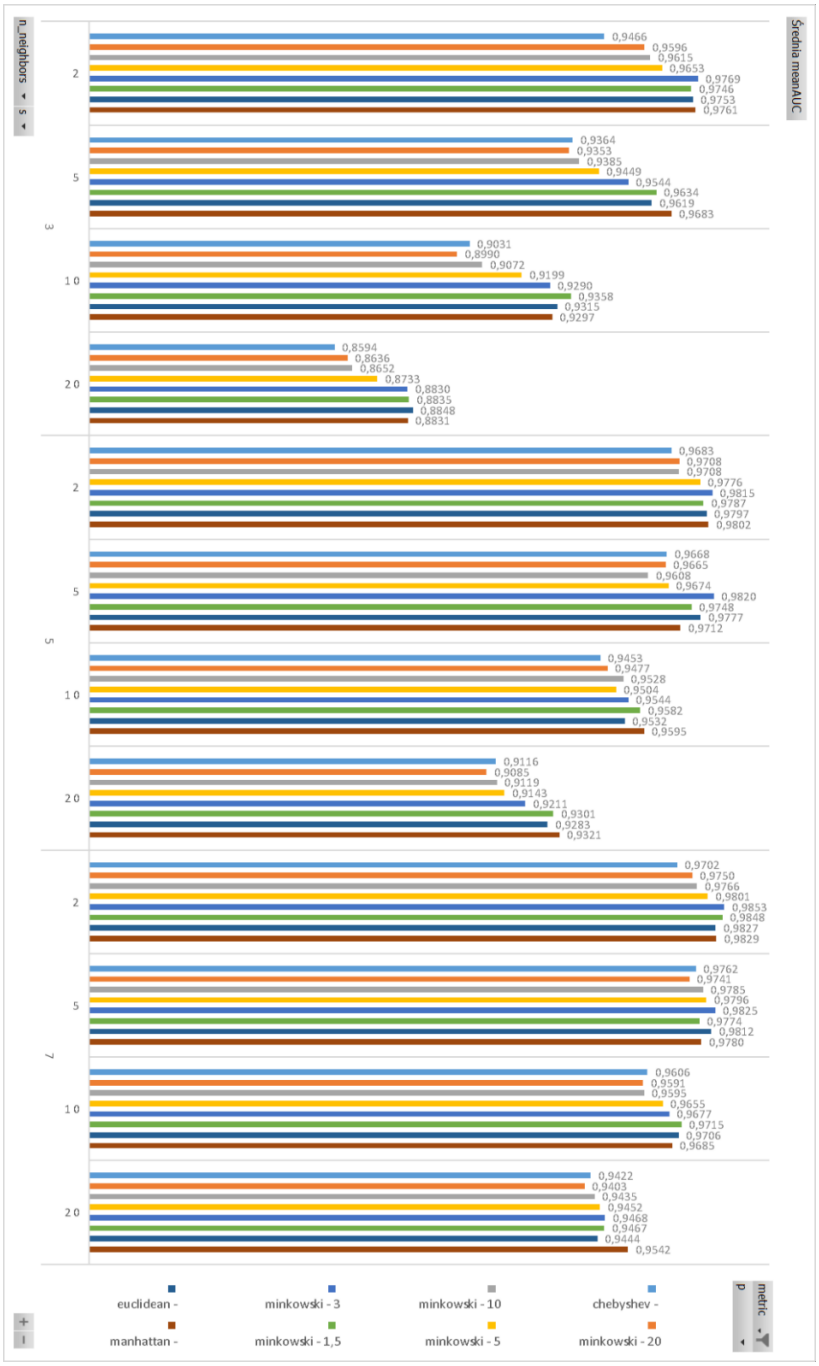
W przeprowadzonym teście otrzymano wartości bliskie 1, które wskazują na brak istotnych różnic między poszczególnymi modelami, dla badanych metryk (niezależnie od liczby sąsiadów i ilości podtabel). Wynik pokazuje, że nie jest zatem konieczne przeprowadzanie dalszych testów w celu wyłonienia najlepszej metryki jako parametru modelu. Dlatego można by stwierdzić, że nie ma jednej najlepszej metryki dla tego modelu. Jednak wydaje się, że są metryki, które warto stosować w podobnych modelach jak metryka Minkowskiego z parametrem *p = 1.5*, metryka Euklidesowa, metryka Manhattan, metryka Canberra, metryka Bray-Curtis. Natomiast są też metryki, których lepiej unikać w podobnych modelach (metryka Chebyszewa oraz metryki Minkowskiego z parametrem *p > 3*).

Przedstawiona będzie teraz analiza jakości działania badanego modelu względem wartości parametru *p* dla rodziny Minkowskiego. Rodzinę Minkowskiego możemy zdefiniować dla wartości parametru *p* większych lub równych niż 1. Dla różnych wartości *p* otrzymujemy różne kule w danej przestrzeni metrycznej, które różnią się kształtem.



Rysunek 12 Kule dla różnych wartości p metryki Minkowskiego źródło [https://en.wikipedia.org/wiki/Minkowski_distance 06.02.2023](https://en.wikipedia.org/wiki/Minkowski_distance)

W przypadku, gdy $p = 1$, otrzymujemy metrykę Manhattan. W przypadku, gdy $p = 2$, otrzymujemy odległość Euklidesową. W przypadku, gdy $p \to \infty$, otrzymujemy odległość Czebyszewa, która jest maksymalną różnicą między współrzędnymi dwóch punktów. Dlatego te szczególne przypadki metryk, które ze względu na swoje nazwy własne zostały oddzielnie zaimplementowane w pracy, w analizie wyników badanego modelu zostały uwzględnione dla rodziny Minkowskiego. Wyniki *AUC* dla rodziny metryk Minkowskiego zostały przedstawione na Rysunku 13 i w Tabeli 8, bez rozróżniania zbiorów danych.



Rysunek 13 Porównanie wyników AUC dla metryk z rodziny Minkowskiego

Tabela 8 Średnie AUC modelu dla metryk z rodziny Minkowskiego

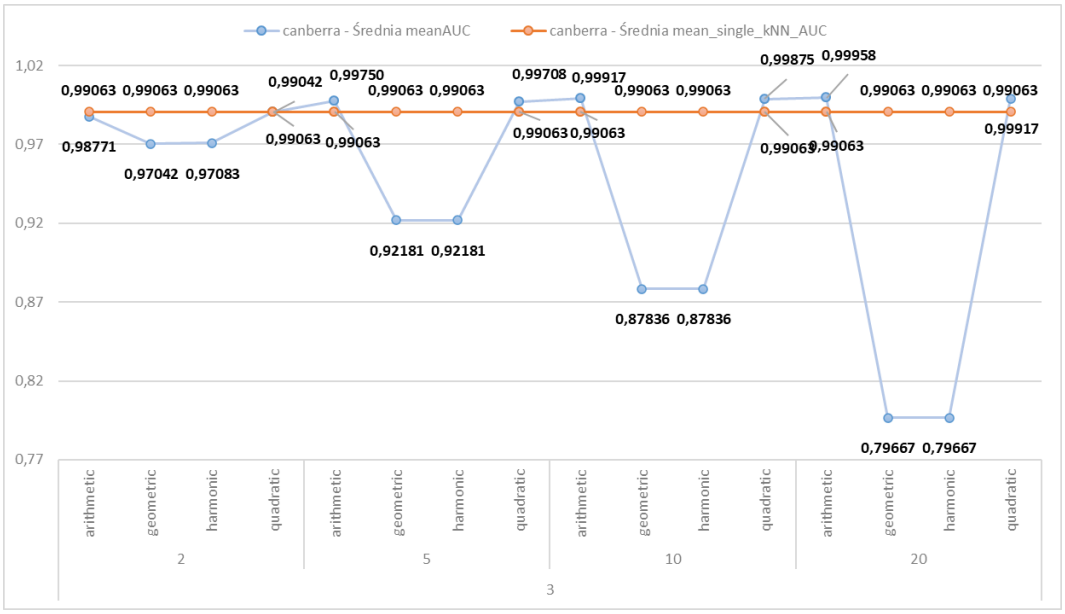
Średnie AUC	Chebyshev	Minkowski p=20	Minkowski p=10	Minkowski p=5	Minkowski p=3	Minkowski p=1,5	Euclidean	Manhattan
k=3								
2	0,9466	0,9596	0,9615	0,9653	**0,9769**	0,9746	0,9753	0,9761
5	0,9364	0,9353	0,9385	0,9449	0,9544	0,9634	0,9619	**0,9683**
10	0,9031	0,8990	0,9072	0,9199	0,9290	**0,9358**	0,9315	0,9297
20	0,8594	0,8636	0,8652	0,8733	0,8830	0,8835	**0,8848**	0,8831
k=5								
2	0,9683	0,9708	0,9708	0,9776	**0,9815**	0,9787	0,9797	0,9802
5	0,9668	0,9665	0,9608	0,9674	**0,9820**	0,9748	0,9777	0,9712
10	0,9453	0,9477	0,9528	0,9504	0,9544	0,9582	0,9532	**0,9595**
20	0,9116	0,9085	0,9119	0,9143	0,9211	0,9301	0,9283	**0,9321**
k=7								
2	0,9702	0,9750	0,9766	0,9801	**0,9853**	0,9848	0,9827	0,9829
5	0,9762	0,9741	0,9785	0,9796	**0,9825**	0,9774	0,9812	0,9780
10	0,9606	0,9591	0,9595	0,9655	0,9677	**0,9715**	0,9706	0,9685
20	0,9422	0,9403	0,9435	0,9452	0,9468	0,9467	0,9444	**0,9542**

W Tabeli 8 pogrubieniem zostały zaznaczone największe wartości AUC w każdym wierszu. Widać, że tylko metryki Minkowskiego z parametrem $p \leq 3$ dawały najlepszą jakość rozważanego modelu. Można by próbować to wyjaśnić kształtem kul odpowiadających danej metryce przedstawionych przykładowo na Rysunku 12 (z jednakowym promieniem). Kula w przestrzeni metrycznej to zbiór punktów tej przestrzeni, których odległość od środka kuli jest mniejsza od danego promienia. Można zauważyć, że wraz ze wzrostem wartości parametru p, wzrasta powierzchniowo obszar kuli oraz kule kolejno zawierają się w sobie. Zatem wyobrażając sobie na płaszczyźnie, że obiekt testowy jest umieszczony w środku kuli, potencjalnie większa ilość obiektów, może znaleźć się w obszarze kuli, co w przypadku kul dla metryk z wysokimi wartościami p (w tym dla metryki Czebyszewa) może powodować konflikty przy wyznaczaniu wartości decyzji dla obiektu testowego. W rezultacie może obniżać to jakość klasyfikacji.

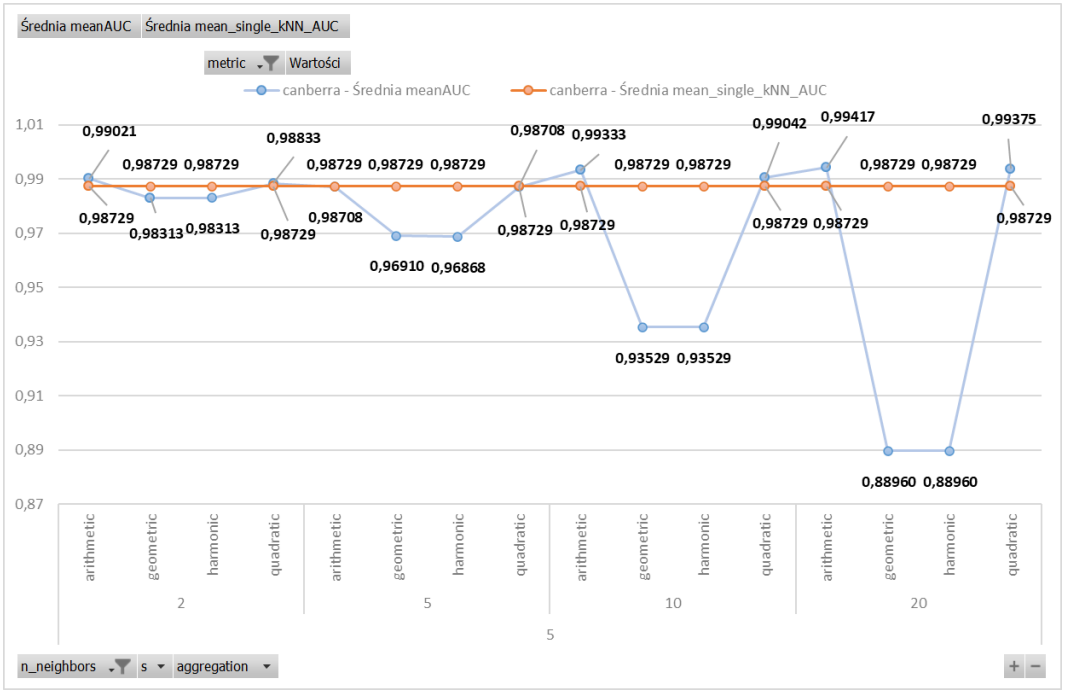
8.3. Analiza jakości modelu względem agregacji i metryk

W tym rozdziale zostanie przeprowadzona analiza wyników badanego modelu głównie względem różnych agregacji i metryk ale także z uwzględnieniem ilości podtabel i sąsiadów. Wykresy powstały na podstawie danych zebranych dla różnych zbiorów danych.

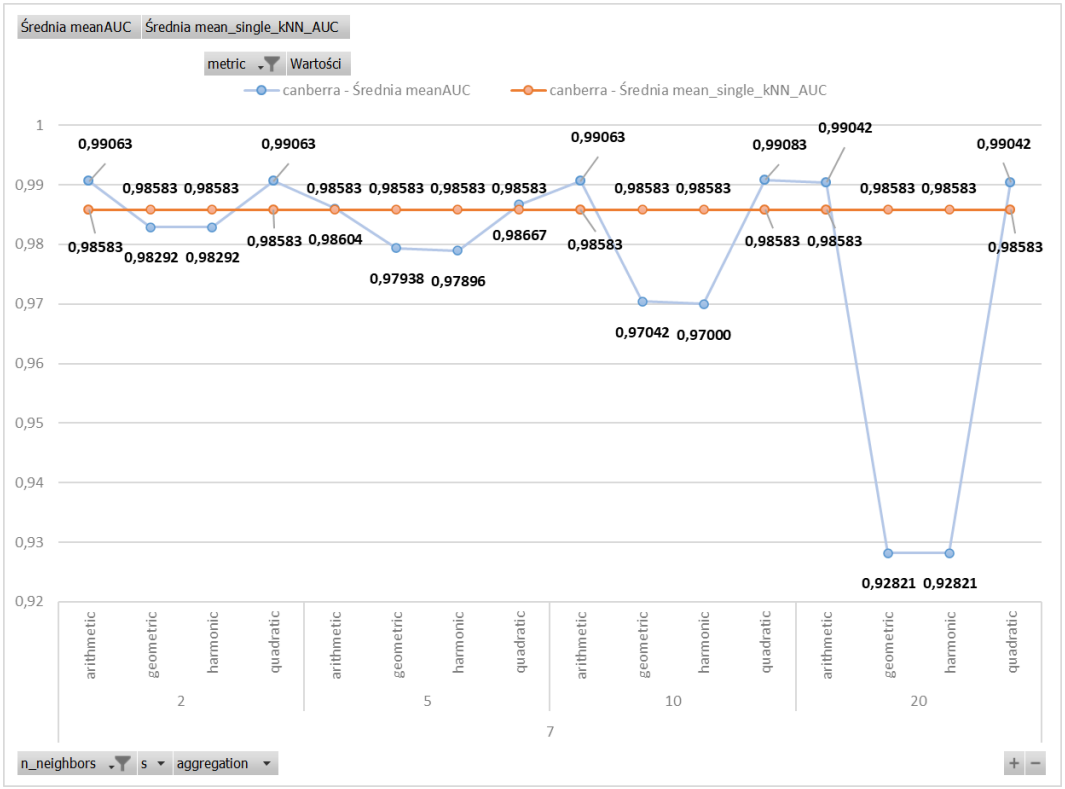
Analiza wyników dla metryki Canberra, dla różnej ilości sąsiadów przedstawiona jest na Rysunkach 14-16.



Rysunek 14 Średnie AUC modelu dla metryki Canberra dla n_neighbors = 3



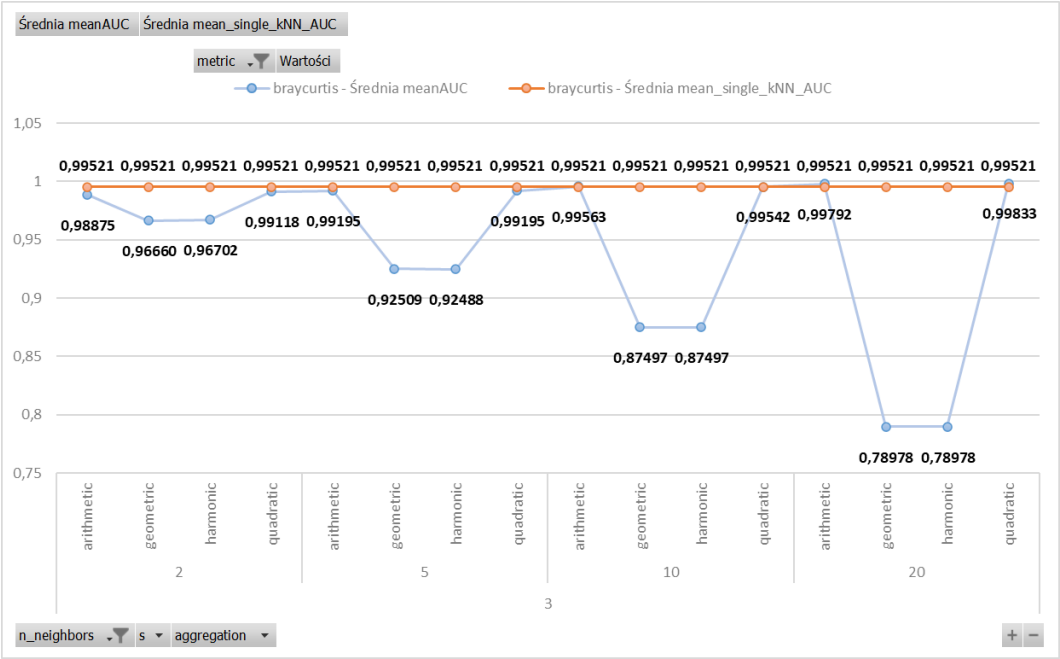
Rysunek 15 Średnie AUC modelu dla metryki Canberra dla n_neighbors = 5



Rysunek 16 Średnie AUC modelu dla metryki Canberra dla n_neighbors = 7

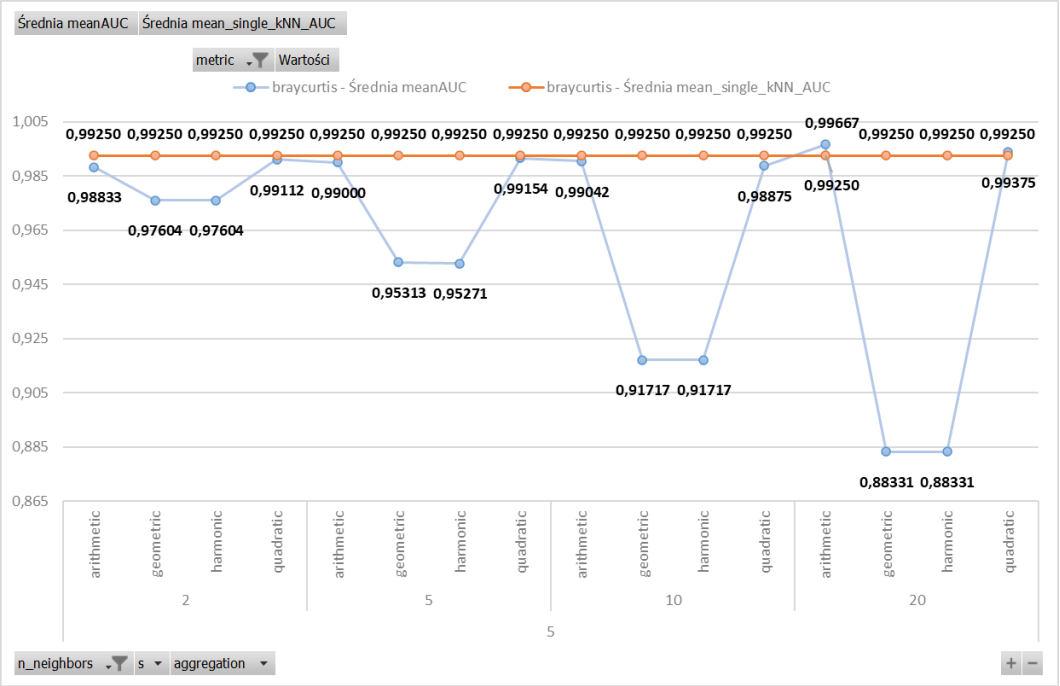
Zastosowanie metryki Canberra pozwala na uzyskanie zadawalającej poprawy jakości klasyfikacji w stosunku do klasyfikatora k-NN przy użyciu odpowiednich agregacji, czyli dla średniej arytmetycznej i kwadratowej. Wynik poprawia się wraz ze wzrostem liczby sąsiadów, jednak czasem nawet bardzo znacząco maleje (w przypadku zastosowania średniej geometrycznej i harmonicznej) wraz ze wzrostem ilości podtabel i najgorszy jest dla ilości podtabel *s=20*.

Podobnie, poniżej przeprowadzona jest analiza wyników dla metryki Bray-Curtis dla różnej ilością sąsiadów a wyniki pokazane są na Rysunkach 17-19.



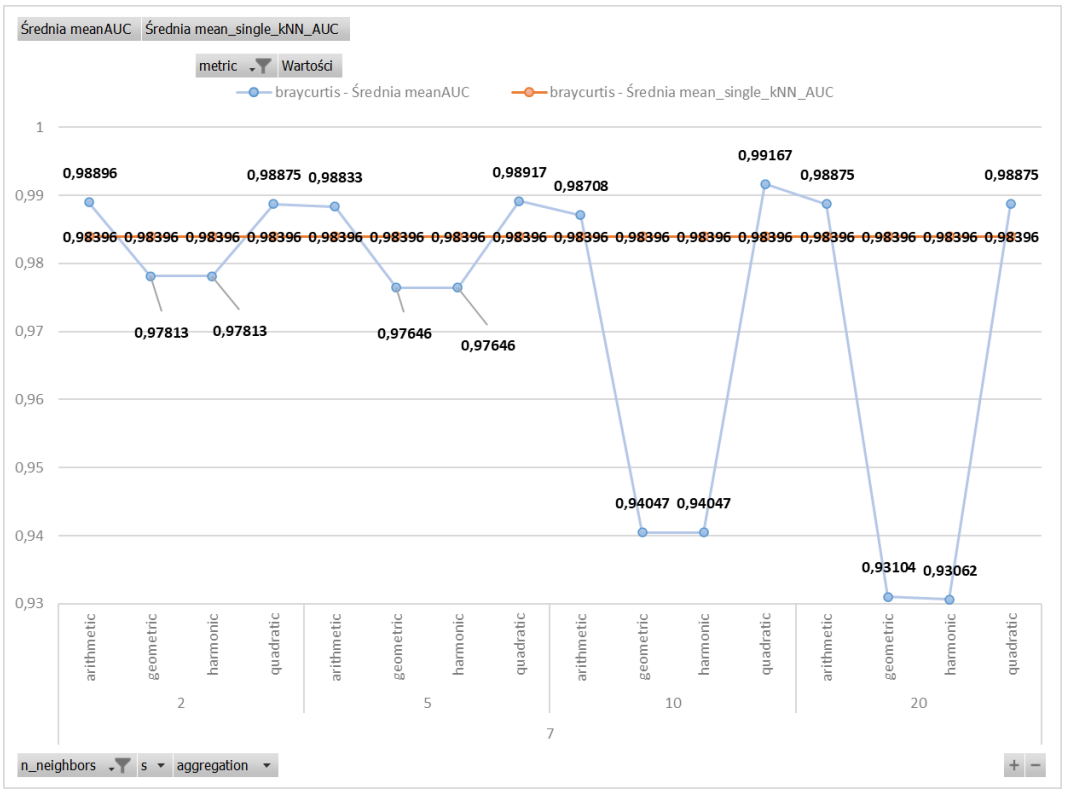
Rysunek 17 Średnie AUC dla metryki Bray-Curtis dla n_neighbors = 3

Na Rysunkach 17 i 18 widzimy, że w przypadku ilości sąsiadów wynoszącej *k=3* lub *k=5* badany model nie polepszył jakości klasyfikacji w porównaniu z odpowiadającym pojedynczym modelem k-NN.



Rysunek 18 Średnie AUC modelu dla metryki Bray-Curtis dla n_neighbors = 5

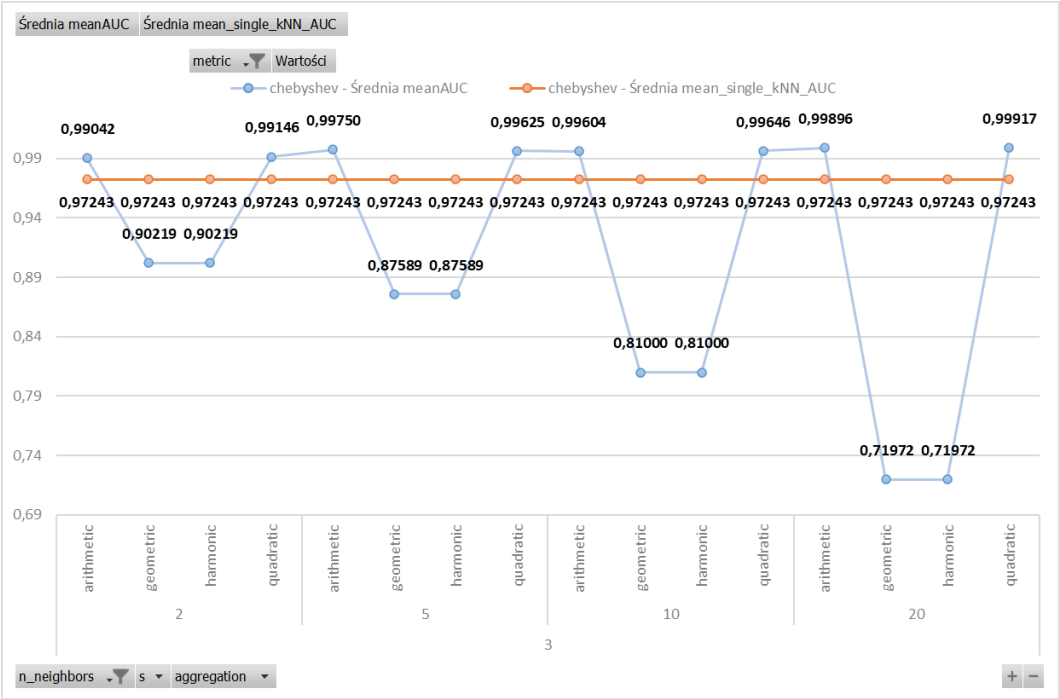
Na Rysunkach 17-19 możemy zaobserwować, że podobnie jak dla metryki Canberra średnie arytmetyczna i kwadratowa dają znacznie lepsze wyniki klasyfikacji w połączeniu z metryką Bray-Curtis niż średnia geometryczna i harmoniczna.



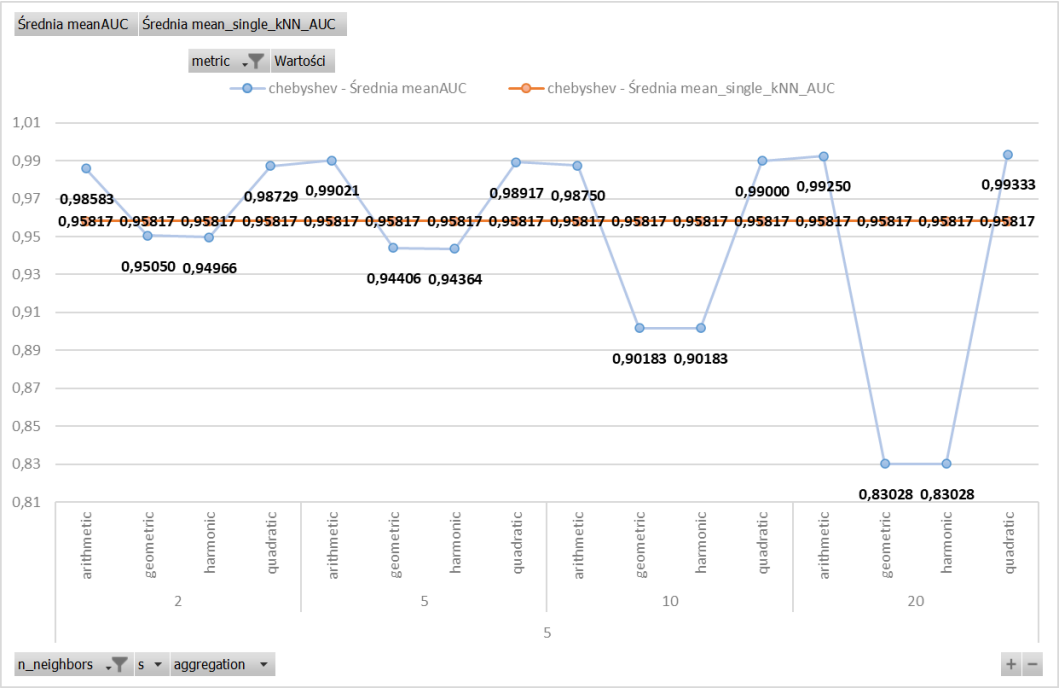
Rysunek 19 Średnie AUC modelu dla metryki Bray-Curtis dla n_neighbors = 7

Na Rysunku 19 widać, że użycie metryki Bray-Curtis przynosi korzyści w porównaniu z zastosowaniem pojedynczego k-NN tylko w przypadku większej liczby sąsiadów (w tym przypadku *k=7*). Na Rysunku 19, dla *s=10* widać też, że zastosowanie średniej kwadratowej może dawać lepsze wyniki klasyfikacji *(AUC=0,99167)* niż zastosowanie średniej arytmetycznej *(AUC=0,98708).*

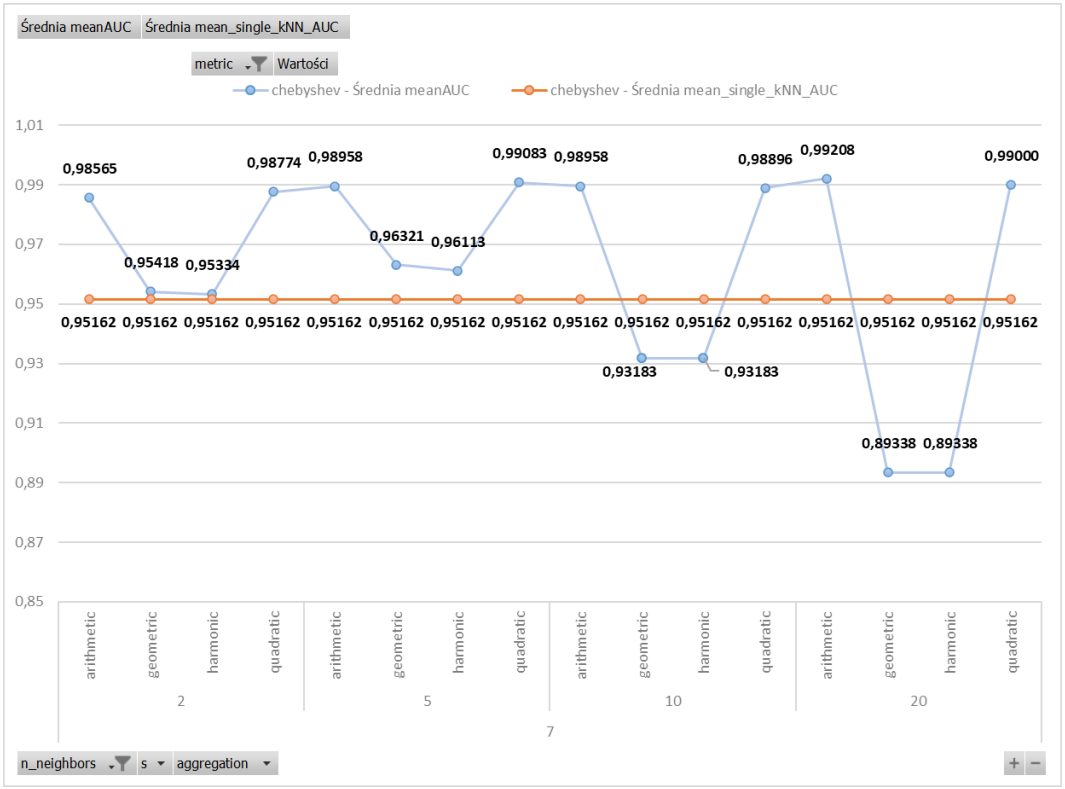
Poniżej znajduje się analiza wyników dla metryki Czebyszewa. Na Rysunkach 20-22 przedstawione są kolejno wyniki dla różnej ilości sąsiadów.



Rysunek 20 Średnie AUC modelu dla metryki Chebyshev dla n_neighbors = 3



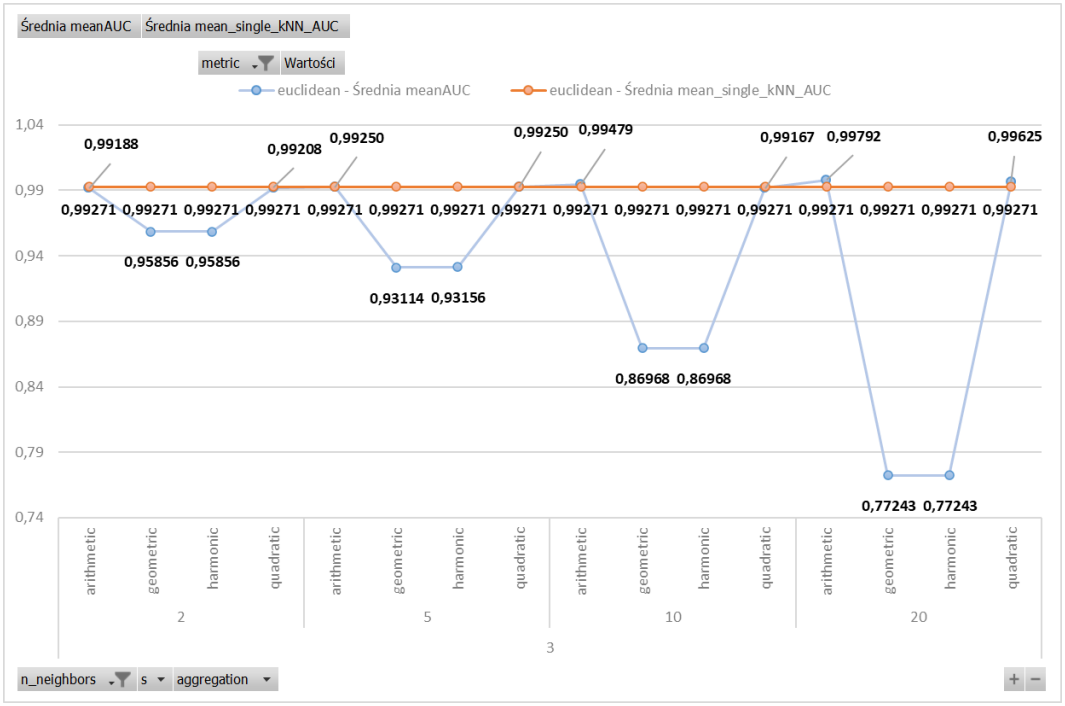
Rysunek 21 Średnie AUC modelu dla metryki Chebyshev dla n_neighbors = 5



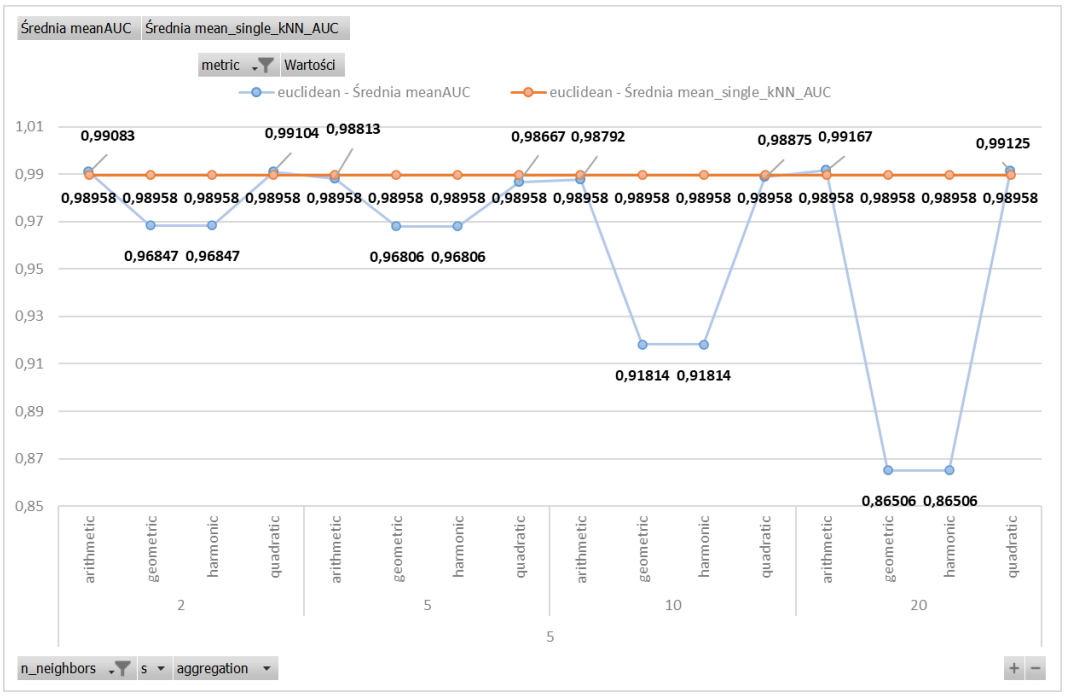
Rysunek 22 Średnie AUC modelu dla metryki Chebyshev dla n_neighbors = 7

Metryka Czebyszewa wyróżnia się na tle innych metryk znacząco polepszając wyniki przy użyciu metody kombinacji klasyfikatorów w porównaniu do pojedynczego k-NN. Wysoka jakość klasyfikacji utrzymuje się już dla *k=3*. Natomiast dla *k=7* jest też inny ciekawy wynik, mianowicie nawet zastosowanie średniej geometrycznej czy harmonicznej, może skutkować (dla odpowiedniej ilości podtabel) polepszeniem jakości klasyfikacji dla badanego modelu w porównaniu z k-NN, co nie miało miejsca w przypadku pozostałych metryk. Zatem metoda kombinacji klasyfikatorów w przypadku najsłabiej sprawującej się metryki okazała się najbardziej znacząca w kontekście poprawy jakości klasyfikacji w porównaniu do pojedynczego k-NN.

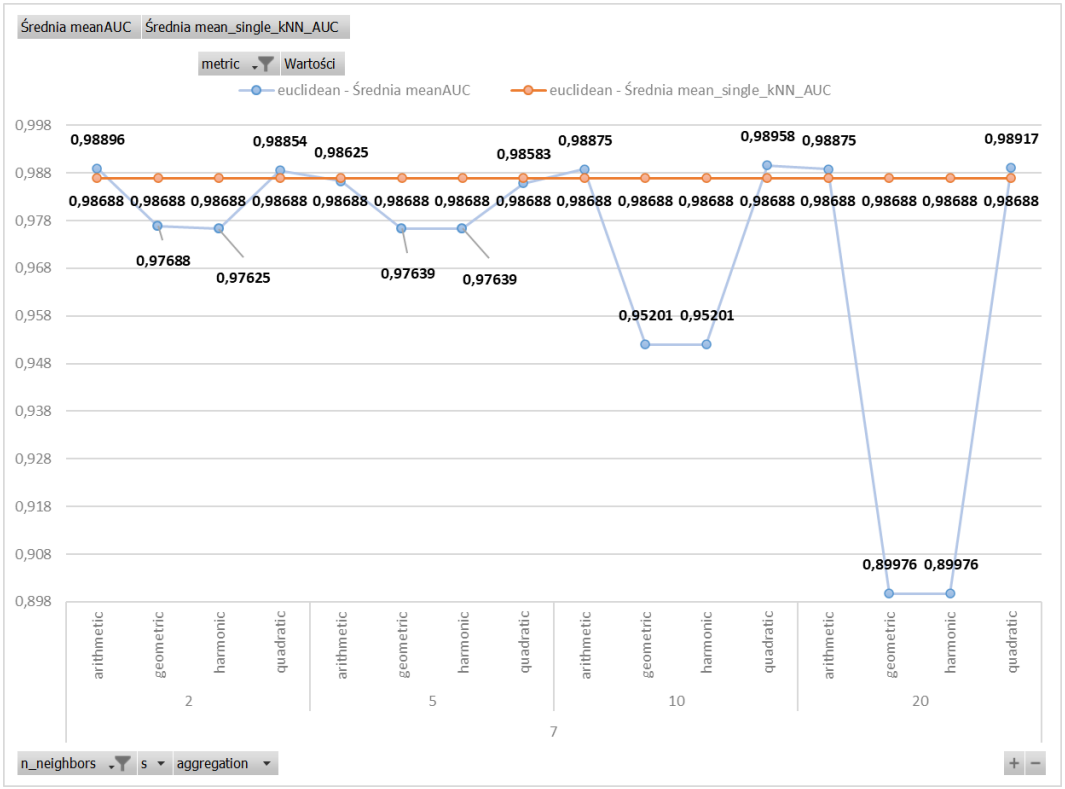
Analiza wyników dla metryki Euklidesowej pokazana jest na Rysunkach 23-25. Widać tutaj podobną tendencję jak dla wcześniejszych metryk – jakość klasyfikacji maleje wraz ze wzrostem ilości podtabel (przy zastosowaniu średniej geometrycznej i harmonicznej).



Rysunek 23 Średnie AUC modelu dla metryki Euclidean dla n_neighbors = 3



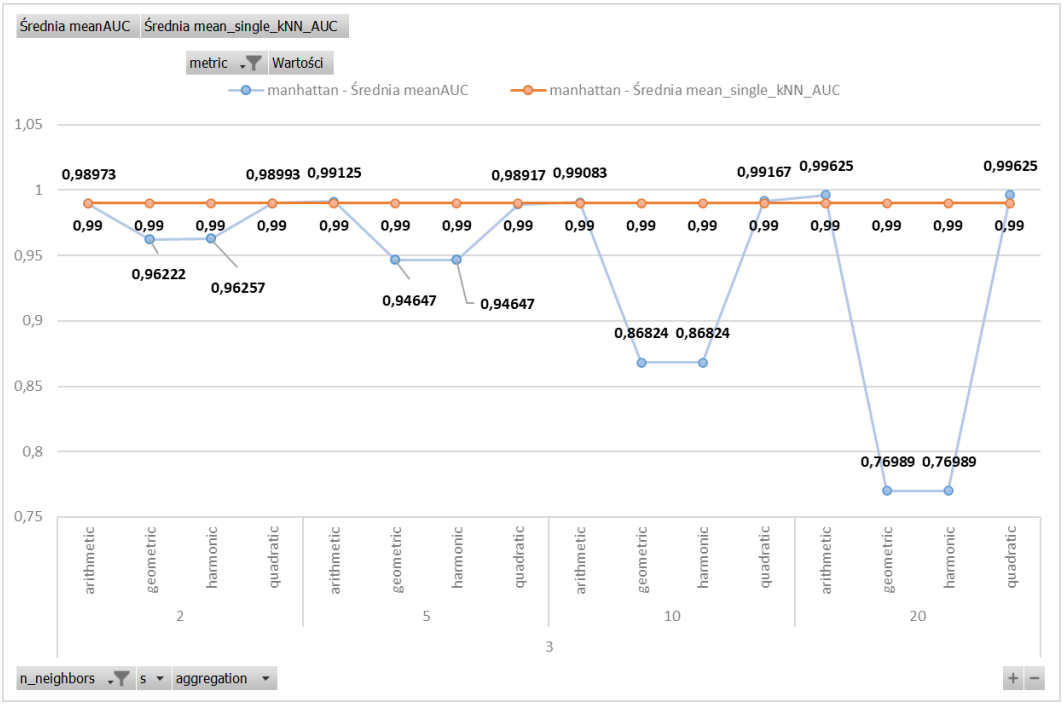
Rysunek 24 Średnie AUC modelu dla metryki Euclidean dla n_neighbors = 5



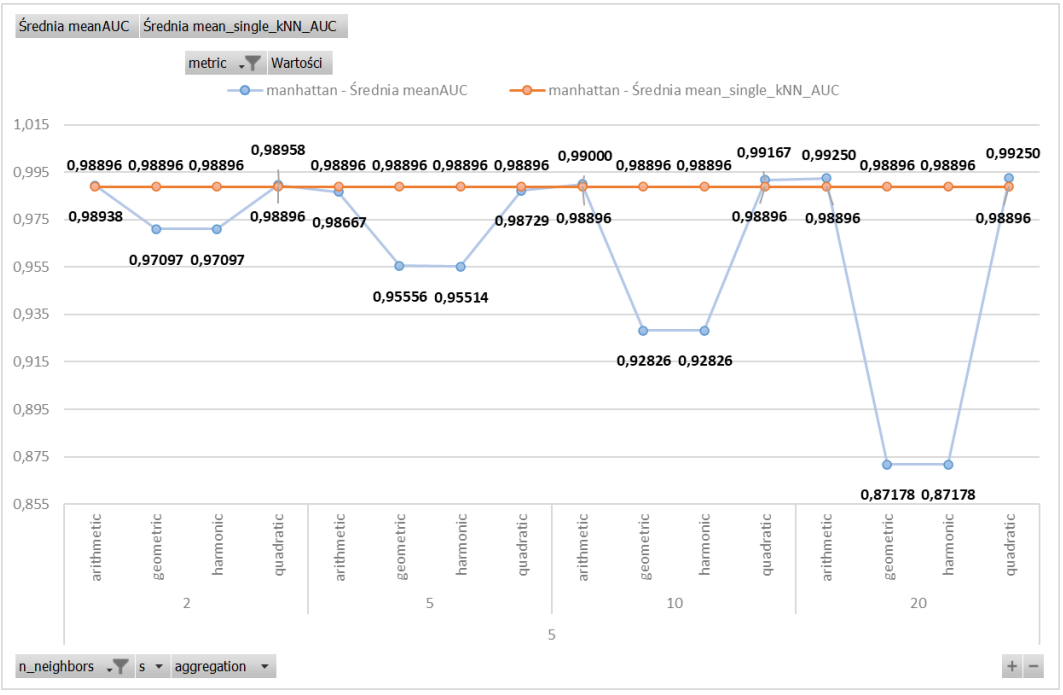
Rysunek 25 Średnie AUC modelu dla metryki Euclidean dla n_neighbors = 7

Dla metryki Euklidesowej, podobnie jak pozostałych metryk, średnie arytmetyczna i kwadratowa dają znacznie lepsze wyniki klasyfikacji niż średnia harmoniczna czy geometryczna. Jednak zastosowana metoda kombinacji klasyfikatorów, jeśli już przewyższa to nieznacznie tylko polepsza jakość klasyfikacji w porównaniu z pojedynczym k-NN.

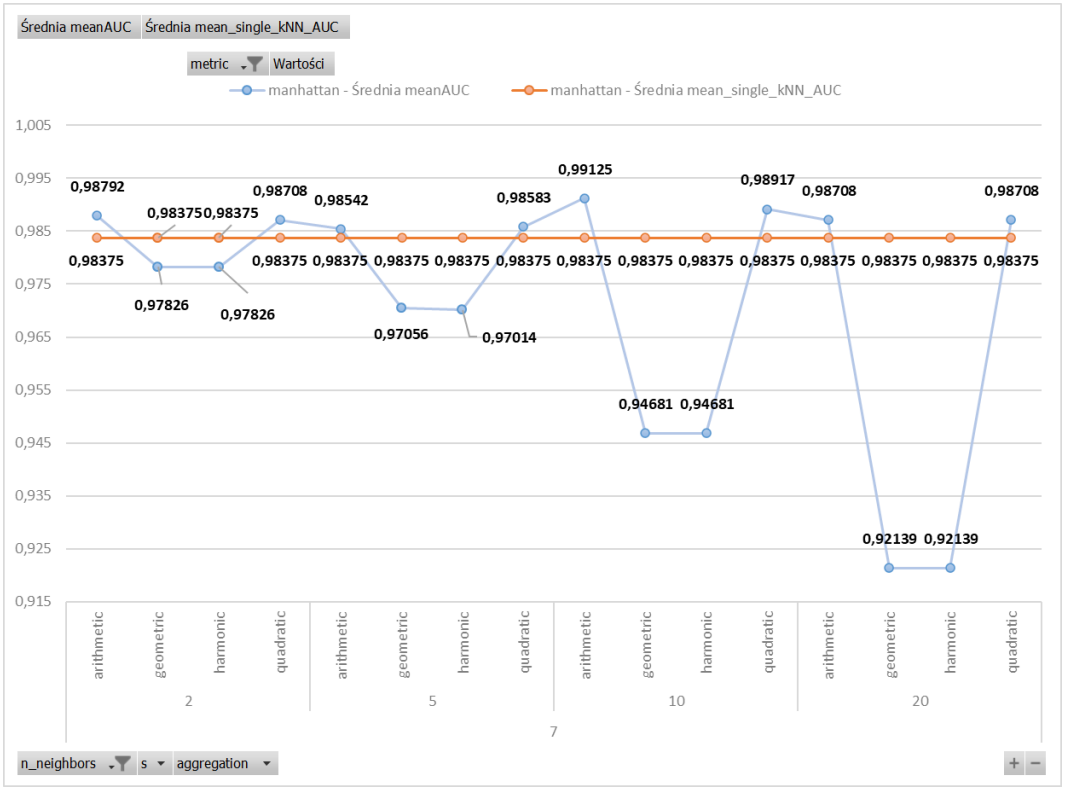
Analiza wyników dla metryki Manhattan przedstawiona jest na Rysunkach 26-28. Rezultaty dotyczące jakości klasyfikacji w zależności od ilości podtabel są analogiczne jak dla pozostałych metryk.



Rysunek 26 Średnie AUC modelu dla metryki Manhattan dla n_neighbors = 3



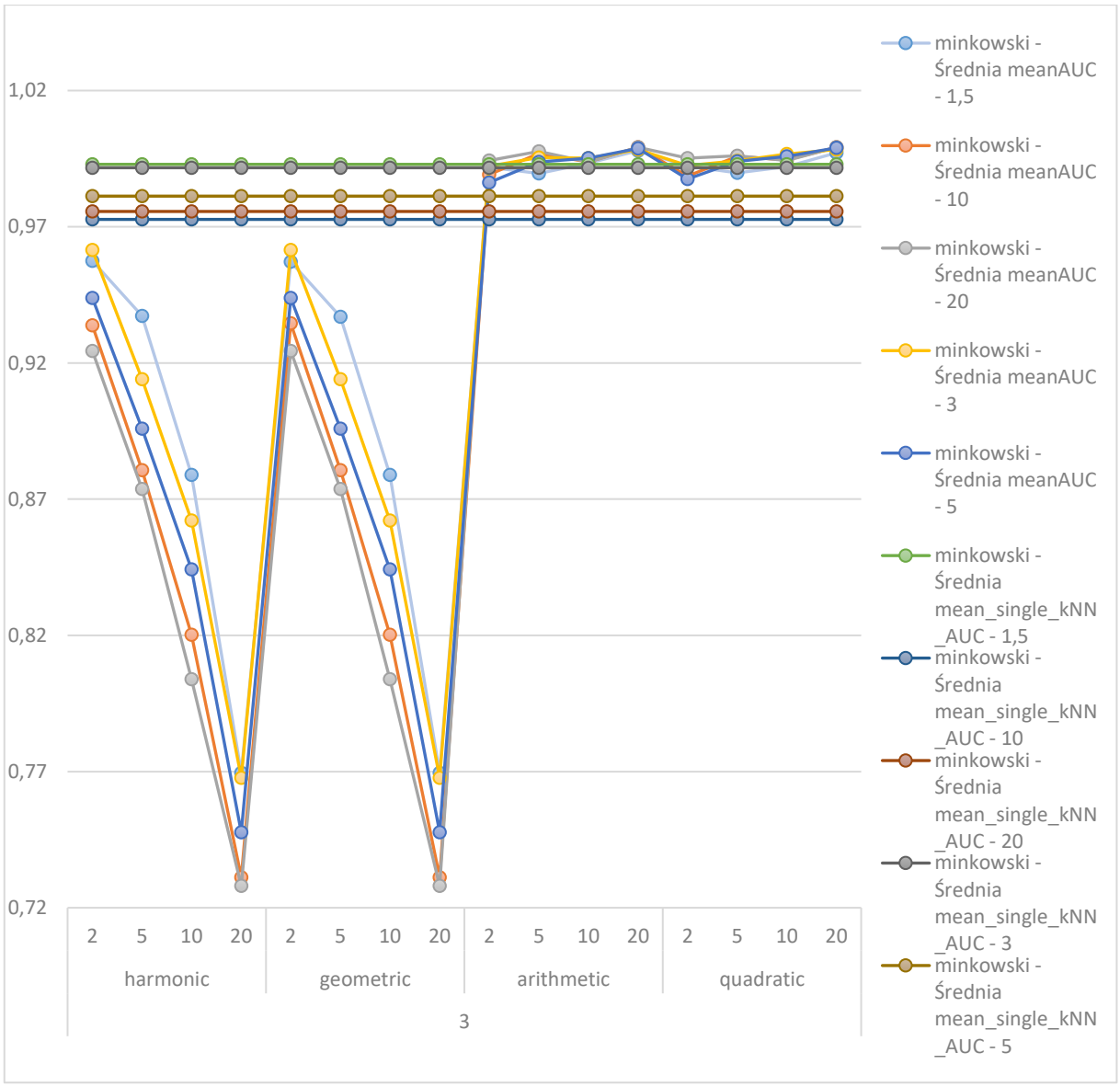
Rysunek 27 Średnie AUC modelu dla metryki Manhattan dla n_neighbors = 5



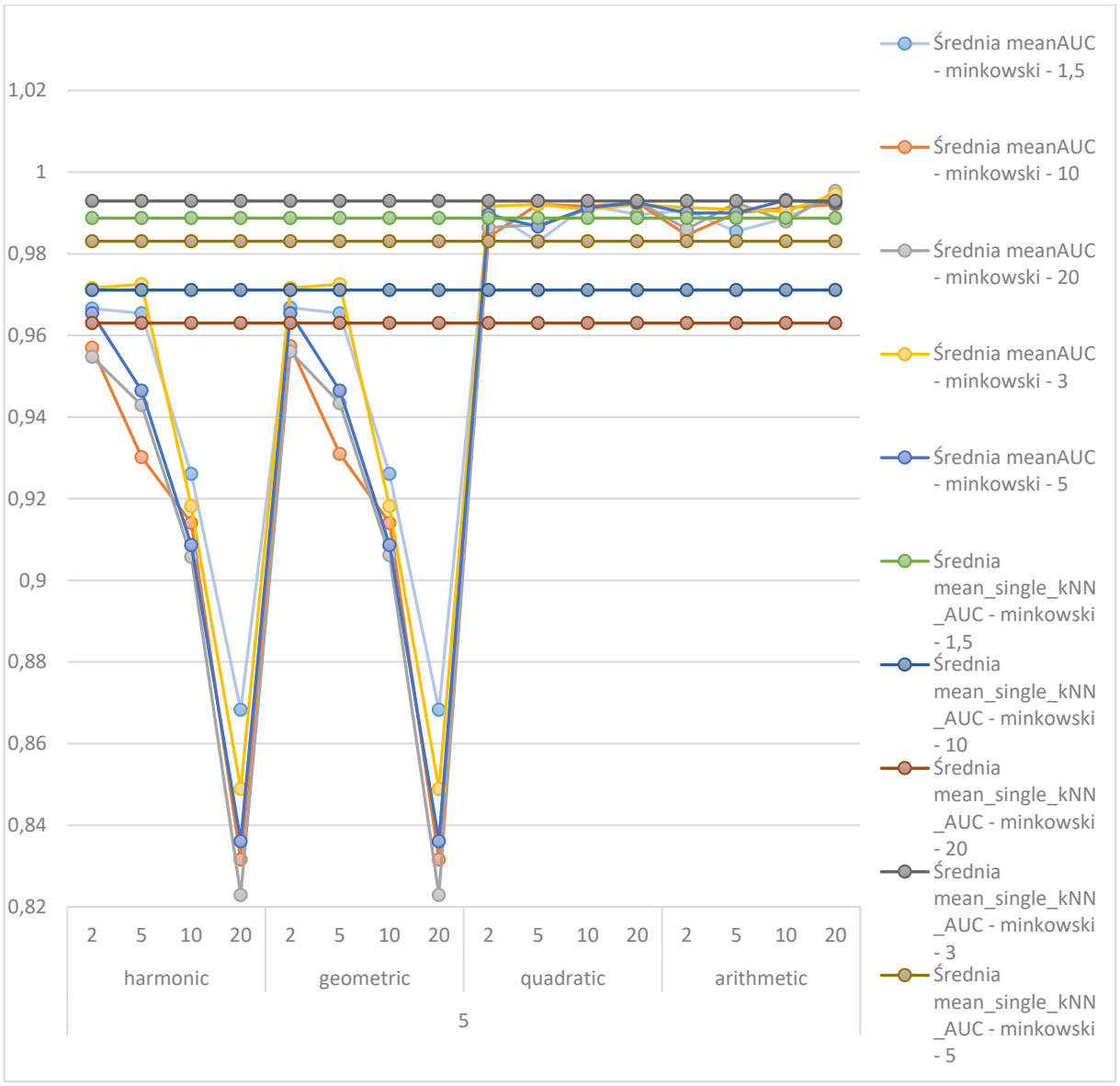
Rysunek 28 Średnie AUC modelu dla metryki Manhattan dla n_neighbors = 7

Metryka Manhattan wnosi wartość dodaną w badanym modelu w porównaniu do pojedynczego k-NN tylko w przypadku większej ilości sąsiadów *k=7*, co widać na Rysunku 28.

Analiza wyników dla metryki Minkowskiego jest przedstawiona na Rysunkach 29-31. Tym razem wykresy (dla różnej ilości sąsiadów) mają inną postać niż w przypadku wcześniej analizowanych metryk, gdyż uwzględniają także różne parametry dla rodziny metryk Minkowskiego.

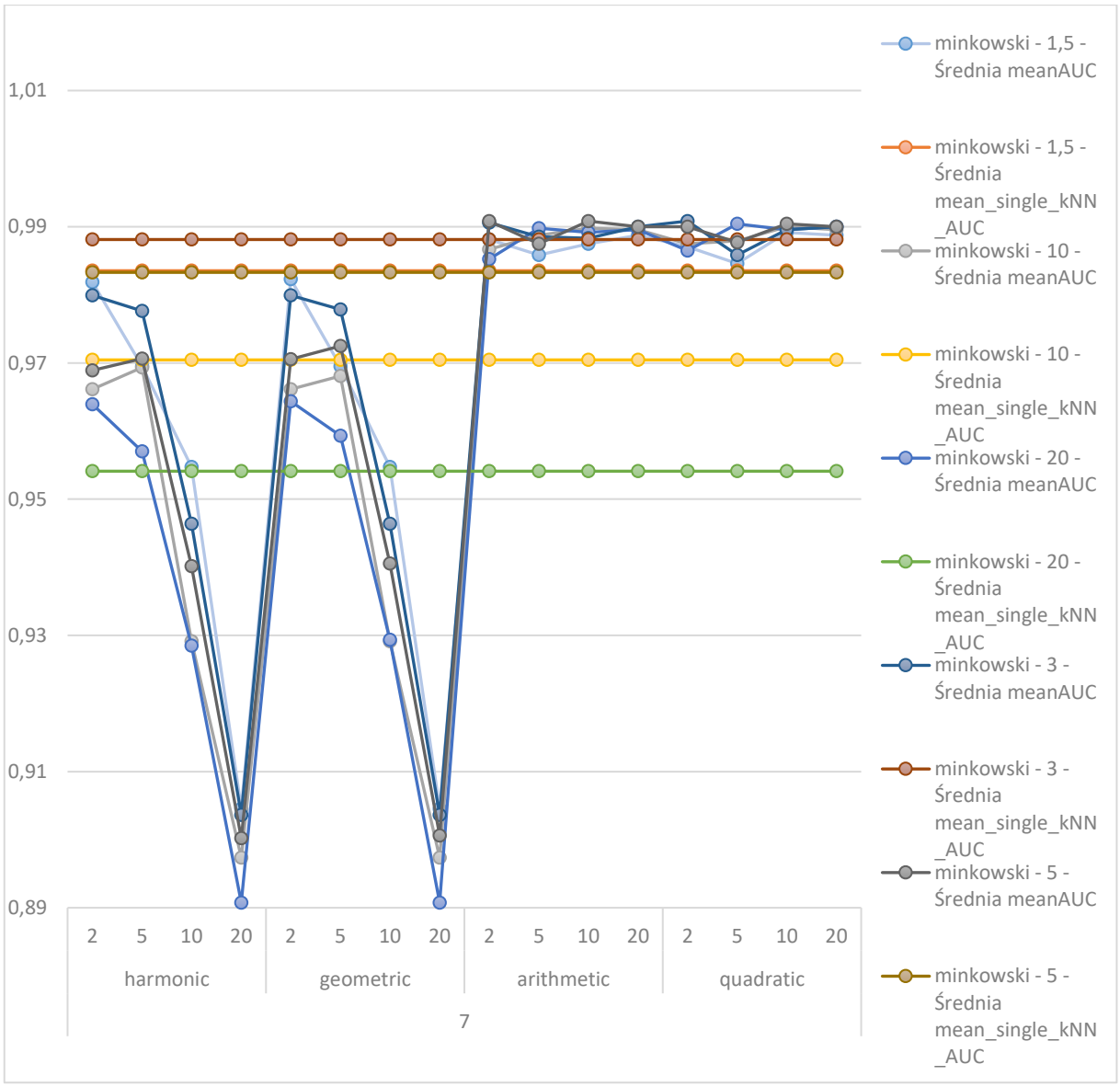


Rysunek 29 Średnie AUC modelu dla metryki Minkowski dla n_neighbors = 3



Rysunek 30 Rysunek 25 Średnie AUC modelu dla metryki Minkowski dla n_neighbors = 5

Na Rysunkach 29-30 widać, że średnie harmoniczna i geometryczna wraz ze wzrostem ilości podtabel dawały znacznie niższą jakość klasyfikacji. Średnie arytmetyczna i kwadratowa dawały dużo bardziej stabilne wyniki klasyfikacji.



Rysunek 31 Rysunek 25 Średnie AUC modelu dla metryki Minkowski dla n_neighbors = 7

Przeglądając wyniki dla metryki Minkowskiego, można zauważyć, że wartość średniej *AUC* rośnie, gdy wartość parametru *p* dla metryki jest mniejsza.

8.4. Wnioski ogólne

Metryka Canberra najczęściej dawała najlepsze wyniki *AUC* w porównaniu z innymi metrykami w modelu łączenia klasyfikatorów, jednak wyniki te nie były istotnie statystycznie lepsze.

Agregacje harmoniczna i geometryczna dają wyniki znacznie gorsze w porównaniu do arytmetycznej i kwadratowej w badanym modelu.

Na wykresach można zauważyć istotną utratę jakości klasyfikacji wraz ze wzrostem podtabel, w szczególności widać to dla słabszych agregacji z parametrem *s=20* (największą rozpatrywaną wartością parametru *s* w tej pracy).

Wraz ze wzrostem liczby sąsiadów można zauważyć poprawę wyników klasyfikacji dla modelu łączenia klasyfikatorów względem pojedynczego k-NN w przypadku użycia agregacji arytmetycznej i kwadratowej.

Najlepsze efekty dla metody łączenia klasyfikatorów uzyskała metryka Czebyszewa w porównaniu z pojedynczym modelem k-NN, chociaż sama w sobie metryka jako parametr modelu dawała najsłabsze wyniki klasyfikacji.

Biorąc pod uwagę analizy przedstawione w rozdziałach 8.2 i 8.3, został ponownie wykonany test istotności różnic pomiędzy badanym modelem, a pojedynczym k-NN (który został pokazany w rozdziale 8.1), ale tym razem dla wybranych parametrów, które powinny w korzystny sposób wpływać na jakość badanego modelu. Zostały zatem wybrane metryki: *Canberra, Manhattan, Czebyszew,* agregacje: *arytmetyczna, kwadratowa*, ilość sąsiadów *k=7* oraz wszystkie badane wartości podtabel *s*. Wyniki testu przedstawione są w Tabeli 9.

Tabela 9 Porównanie jakości modelu dla wybranych parametrów z pojedynczym klasyfikatorem k-NN

Zbiór danych	Średnia AUC kombinacja klasyfikatorów	Średnia AUC pojedynczy k-NN	Odchylenie standardowe kombinacja klasyfikatorów	Odchylenie standardowe pojedynczy k-NN
Colon	0,96007	0,90903	0,00899	0,04182
Prostate	0,98400	0,96642	0,00648	0,02692
Leukemia	1	0,99333	0	0,00943
Ovarian	1	0,99988	0	0,00016
Lymphoma	1	1	0	0

Wyniki testu Manna-Whitneya dla poszczególnych zestawów danych przedstawiają się następująco: *Ovarian: p = 0,00671665, Leukemia: p = 0,00671665, Colon: p = 0,00580485, Prostate: p = 0,45861435, Lymphoma: p = 1.*

Można zauważyć, że dobór odpowiednich parametrów jest kluczowy dla osiągnięcia istotnie statystycznie lepszych wyników klasyfikacji. W tym przypadku, model kombinacji klasyfikatorów miał istotnie wyższe wartości średniej *AUC* dla zestawów danych *Ovarian,*

Leukemia i Colon, co sugeruje, że w tych przypadkach kombinacja klasyfikatorów była bardziej skuteczna niż pojedynczy k-NN. Jednakże, dla zestawów danych *Prostate* i *Lymphoma* wyniki wskazują, że nie ma statystycznie istotnej różnicy między modelami. Wybór wspomnianych wcześniej parametrów pozytywnie wpłynął na jakość klasyfikacji i stabilność wyników modelu kombinacji klasyfikatorów, co widać porównując odchylenie standardowe z Tabeli 4 oraz Tabeli 9. Zatem dla trzech zbiorów danych udało się istotnie polepszyć jakość badanego modelu poprzez dobór odpowiednich parametrów. Dla zbioru danych *Lymphoma* nie było to możliwe, ze względu na wysoką jakość klasyfikatora bazowego, *AUC=1*.

9. Podsumowanie

Celem tej pracy było przeprowadzenie analizy porównawczej różnych metryk i agregacji na danych mikromacierzowych, aby sprawdzić ich wpływ na jakość rozpatrywanego modelu uczenia maszynowego. Cel ten został osiągnięty i pokazano, że najlepsze rezultaty osiągnęły metryki Minkowskiego z parametrem bliższym 1 oraz metryka Canberra. Nie osiągnięto jednak istotnie lepszych rezultatów klasyfikacji dla tych metryk jako parametrów rozpatrywanego modelu, co zostało wykazane testami statystycznymi. Jednak wyniki dla metryki Canberra, która najczęściej dawała najwyższe wyniki klasyfikacji wskazują na to, że warto stosować ją w algorytmach podobnych do rozważanego w tej pracy. Agregacje, które okazały się najlepsze dla rozpatrywanego modelu to średnia arytmetyczna i kwadratowa z porównywalnymi wynikami. Ponadto odnotowana została utrata jakości klasyfikacji wraz ze wzrostem ilości agregowanych modeli dla średniej geometrycznej i harmonicznej (wartość $s=20$), więc nasuwa się wniosek, że w badanym modelu należałoby łączyć mniejszą ilość modeli bazowych, biorąc pod uwagę różne agregacje, ale może to też mieć związek z ilością wyselekcjonowanych cech przez algorytm RFECV, które są następnie przydzielane do podtabel. Dodatkowo, wraz ze wzrostem liczby sąsiadów można zauważyć poprawę wyników klasyfikacji dla badanego modelu łączenia klasyfikatorów.

W dalszych badaniach nad rozpatrywanym modelem można oczywiście też rozpatrywać inne metryki, które znane są w literaturze [1] oraz inne rodzaje średnich czy też ogólniej funkcji agregacji.

10. Literatura

[1] H.A. Abu Alfeilat, A.B.A. Hassanat, O. Lasassmeh, A.S. Tarawneh, M.B. Alhasanat, H.S. Eyal Salman, V.B.S. Prasath, Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review, Big Data, 7(4) (2019) 221-248.

[2] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, J.M. Benítez, F. Herrera, A review of microarray datasets and applied feature selection methods, Information Sciences, 282 (2014) 111-135.

[3] T. Morzy, Eksploracja danych. Metody i algorytmy, PWN, Warszawa 2013.

[14] G. Wojcik, M. Wazny, Bray-curtis metrics as measure of liquid state machine separation ability in function of connections density, Procedia Computer Science, 51 (2015) 2979-2983.

[16] P.S. Bullen, D.S, Mitrinović, P.M. Vasić, Means and Their Inegualities, Reidel, Dordrecht 1988.

[17] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers 1998.

[18] S. Suthaharan, Machine Learning Models and Algorithms for Big Data Classification, Springer Boston MA, 2016, 207-235

[21] N. Nagaraj, B.M. Vikranth, N. Yogesh, Recursive Feature Elimination Technique for Technical Indicators Selection, in: A. Bennour, T. Ensari, Y. Kessentini, S. Eom (eds), Intelligent Systems and Pattern Recognition. ISPR 2022. Communications in Computer and Information Science, vol 1589. Springer, Cham, 2022, pp. 139-145.

[29] A. Snarska, Statystyka Ekonometria Prognozowanie Ćwiczenia z Excelem, Agencja Wydawnicza Placet 2007.

11. Netografia

[4] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html 16.02.2023

[5] https://pl.wikipedia.org/wiki/Przestrze%C5%84_metryczna 16.02.2023

[6] https://en.wikipedia.org/wiki/Minkowski_distance 16.02.2023

[7] https://pl.wikipedia.org/wiki/Przestrze%C5%84_Lp 16.02.2023

[8] https://en.wikipedia.org/wiki/Euclidean_distance 16.02.2023

[9] https://en.wikipedia.org/wiki/Taxicab_geometry 16.02.2023

[10] http://vistula.pk.edu.pl/~qmq/SAD/Analiza_skupien.pdf 16.02.2023

[11] https://pl.wikipedia.org/wiki/Odleg%C5%82o%C5%9B%C4%87_Czebyszewa 16.02.2023

[12] https://people.revoledu.com/kardi/tutorial/Similarity/CanberraDistance.html 16.02.2023

[13] https://people.revoledu.com/kardi/tutorial/Similarity/BrayCurtisDistance.html 16.02.2023

[15] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.DistanceMetric.html 16.02.2023

[19] https://www.educba.com/support-vector-regression/ 16.02.2023

[20] https://towardsdatascience.com/support-vector-regression-svr-one-of-the-most-flexible-yet-robust-prediction-algorithms-4d25fbdaca60 16.02.2023

[22] https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html 16.02.2023

[23] https://leo.ugr.es/elvira/DBCRepository/ 16.02.2023

[24] https://leo.ugr.es/elvira/DBCRepository/ColonTumor/ColonTumor.html 16.02.2023

[25] https://leo.ugr.es/elvira/DBCRepository/DLBCL/DLBCL-Stanford.html 16.02.2023

[26] https://leo.ugr.es/elvira/DBCRepository/Leukemia/ALLAML.html 16.02.2023

[27] https://leo.ugr.es/elvira/DBCRepository/OvarianCancer/OvarianCancer-NCI-PBSII.html 16.02.2023

[28] https://leo.ugr.es/elvira/DBCRepository/ProstateCancer/ProstateCancer.html 16.02.2023

[30] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.mannwhitneyu.html 16.02.2023

[31] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kruskal.html 16.02.2023

13. Wzory

(1) funkcja określająca metrykę

(2) własność identyczności nierozróżnialnych dla przestrzeni metrycznej

(3) własność symetrii dla przestrzeni metrycznej

(4) własność nierówności trójkąta dla przestrzeni metrycznej

(5) metryka Minkowskiego

(6) nierówność Minkowskiego

(7) metryka Euclidesa

(8) metryka Manhattan

(9) metryka Czebyszewa

(10) metryka Canberra

(11) metryka Bray-Curtis

(12) warunek średniej

(13) średnia arytmetyczna

(14) średnia geometryczna

(15) średnia harmoniczna

(16) średnia kwadratowa

14. Spis rysunków

15. Spis tabel

16. Streszczenie

Celem pracy jest zbadanie i porównanie skuteczności zastosowania różnych metryk i sposobów agregacji w algorytmie kombinacji klasyfikatorów *k* najbliższych sąsiadów (k-NN) w przypadku dużych zbiorów danych, szczególnie w odniesieniu do mikromacierzy DNA wykorzystywanych w badaniach genetycznych.

W pracy skupiono się na analizie wpływu różnych parametrów algorytmu k-NN na skuteczność klasyfikacji, w tym na wybór metryki i liczby sąsiadów. Literatura wskazuje, że najczęściej stosowanymi metrykami w algorytmach k-NN są metryki Euklidesowa i Manhattan, dlatego w tej pracy zbadano również inne metryki Minkowskiego z różnymi parametrami oraz metryki Canberra i Bray-Curtis. Ważnym elementem pracy jest też porównanie różnych sposobów agregacji wyników klasyfikacji uzyskanych przez poszczególne klasyfikatory uczące się na podzbiorach danych. W tym celu zbadano kilka znanych agregacji, mianowicie średnią arytmetyczną, kwadratową, geometryczną i harmoniczną.

Dodatkowo, zastosowano metodę selekcji cech Recursive Feature Elimination with Cross-Validation (RFECV), która wykorzystuje estymator do iteracyjnego wyboru najbardziej istotnych cech w danych. Aby potencjalnie zwiększyć skuteczność klasyfikacji, atrybuty wybrane przez RFECV są dzielone na mniejsze podzbiory, a klasyfikatory są uczone na każdym z nich.

W badaniach wykorzystano język programowania Python oraz biblioteki *scikit-learn i scipy.stats*, które oferują gotowe implementacje algorytmu k-NN, różnych metryk oraz metod agregacji. Przeprowadzone eksperymenty pozwoliły na znalezienie optymalnych wartości parametrów badanego modelu łączenia klasyfikatorów k-NN oraz na wybór najlepszych agregacji. Przeprowadzone analizy mogą pomóc w doborze parametrów rozważanego modelu, a w konsekwencji poprawić jakość metod uczenia maszynowego w kontekście danych mikromacierzowych.

The aim of the work is to investigate and compare the effectiveness of different metrics and aggregation methods in the ensemble model of k-nearest neighbors (k-NN) algorithm for large data sets, particularly in relation to DNA microarrays used in genetic research. The focus of the study is on the analysis of the impact of various k-NN algorithm parameters on classification effectiveness, including the choice of metric and number of neighbors. The literature suggests that the most commonly used metrics in k-NN algorithms are the Euclidean and Manhattan metrics, so this work also examines other Minkowski metrics with different parameters, as well as the Canberra and Bray-Curtis metrics. An important aspect of the work is also the comparison of different methods of aggregating classification results obtained by individual classifiers trained on subsets of data. To this end, several well-known aggregations were investigated, namely arithmetic, quadratic, geometric, and harmonic means.

Additionally, the Recursive Feature Elimination with Cross-Validation (RFECV) feature selection method was applied, which uses an estimator to iteratively select the most important features in the data set. To increase potentially classification effectiveness, the attributes selected by RFECV are divided into smaller subsets, and classifiers are trained on each of them.

Python programming language and libraries such as scikit-learn and scipy.stats, which provides ready-made implementations of the k-NN algorithm, various metrics, and aggregation methods, were used in the research. The conducted experiments allowed for finding optimal parameter values for the examined k-NN classifier model and selecting the best aggregations. The analyses carried out can help in selecting parameters for the considered model and ultimately improve the quality of machine learning methods in the context of microarray data.

Załącznik nr 2 do Zarządzenia nr 228/2021 Rektora Uniwersytetu Rzeszowskiego z dnia 1 grudnia 2021 roku w sprawie ustalenia procedury antyplagiatowej w Uniwersytecie Rzeszowskim

OŚWIADCZENIE STUDENTA O SAMODZIELNOŚCI PRACY

Paweł Maciej Durda
Imię (imiona) i nazwisko studenta

Kolegium Nauk Przyrodniczych

Informatyka
Nazwa kierunku

096449
Numer albumu:

1. Oświadczam, że moja praca dyplomowa pt.: . Analiza metryk i agregacji w zagadnieniach kombinacji klasyfikatora k-NN w przypadku dużej liczby atrybutów w zbiorach danych

   1) została przygotowana przeze mnie samodzielnie*,
   2) nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2021 r., poz. 1062) oraz dóbr osobistych chronionych prawem cywilnym,
   3) nie zawiera danych i informacji, które uzyskałem w sposób niedozwolony,
   4) nie była podstawą nadania dyplomu uczelni wyższej ani mnie, ani innej osobie.

2. Jednocześnie *wyrażam zgodę/ nie wyrażam zgody* ** na udostępnienie mojej pracy dyplomowej do celów naukowo-badawczych z poszanowaniem przepisów ustawy o prawie autorskim i prawach pokrewnych.

....
(miejscowość, data) *(czytelny podpis studenta)*

***** *uwzględniając merytoryczny wkład promotora pracy*
*** - niepotrzebne skreślić*