Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH

Instytut Informatyki
Zakład Systemów Informacyjnych

Praca dyplomowa magisterska

na kierunku INFORMATYKA
w specjalności Inżynieria Systemów Informatycznych

Meta-metody służące do poprawy jakości klasyfikacji

Konrad Ziaja
nr albumu 272170

promotor
dr inż. Łukasz Skonieczny

Warszawa 2017

Warszawa, 1 marca 2017

POLITECHNIKA WARSZAWSKA
Wydział Elektroniki i Technik Informacyjnych

OŚWIADCZENIE

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa magisterska pt. Meta-metody służące do poprawy jakości klasyfikacji:

- została napisana przeze mnie samodzielnie,
- nie narusza niczyich praw autorskich,
- nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam, że przedłożona do obrony praca dyplomowa nie była wcześniej podstawą postępowania związanego z uzyskaniem dyplomu lub tytułu zawodowego w uczelni wyższej. Jestem świadom, że praca zawiera również rezultaty stanowiące własności intelektualne Politechniki Warszawskiej, które nie mogą być udostępniane innym osobom i instytucjom bez zgody Władz Wydziału Elektrycznego.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Konrad Ziaja.....

Spis treści

Wstęp

Ostatnie dekady to wielki skok cywilizacyjny. Nastąpił gwałtowny rozwój informatyki oraz pokrewnych dziedzin. Wskutek postępu technicznego, ludzie zaczęli generować ogromne ilości danych, które zostają zapisane na co raz większych i tańszych nośnikach danych. Codziennie, podczas zwykłych czynności np. płacenia kartą w sklepie, przeglądania internetu, wysyłania poczty e-mail, nosząc ze sobą podłączony do sieci GSM telefon tworzymy nieświadomie duże ilości danych. Firmy ubezpieczeniowe, banki, sklepy, sieci handlowe i inne instytucje prywatne oraz publiczne zbierają i przetwarzają o nas różne informacje, a następnie gromadzą je w ogromnych bazach danych. Dziedzinami zajmującymi się przetwarzaniem danych oraz wyszukiwaniem wzorców i wiedzy w danych jest uczenie maszynowe (ang. *machine learning*) oraz eksploracja danych (ang. *data mining*). Uczenie maszynowe można podzielić między innymi na uczenie nadzorowane (ang. *supervised learning*) oraz uczenie nienadzorowane (ang. *unsupervised learning*). W uczeniu nadzorowanym program dla znanych danych z przypisanymi klasami lub kategoriami tworzy reguły decyzyjne, które następnie wykorzystywane są do przypisania klasy nowym nieznanym wcześniej danym. Natomiast w uczeniu nienadzorowanym, wyszukiwane są wzorce, struktury w danych bez przypisanej końcowej kategorii.

Elementem uczenia nadzorowanego jest klasyfikacja danych (ang. *classification*). W procesie klasyfikacji danych, system wyszukuje wzorców oraz reguł w danych uczących ze znanymi kategoriami, a następnie przypisuje kategorię nowym obserwacjom bez określonej klasy. Oczywiście przypisana kategoria pochodzi ze zbioru danych uczących. Przykładem klasyfikacji danych może być proces przyznania kredytu. Bank zbiera różne informacje o klientach (rodzaj pracy, umowy, wysokość zarobków, poprzednie pożyczki, kredyty, terminowość spłat rat), a następnie po jakimś czasie oznacza czy był to dobry klient z korzyścią dla banku. Klasyfikator nauczony tymi danymi, określa czy można udzielić kredytu nowemu klientowi na podstawie dostarczonych danych. Innym przykładem mogą być dane medyczne np. osób chorych na raka. Lekarz gromadzi parametry medyczne pacjentów wraz z kategorią czy dana osoba jest chora. Na podstawie tych danych, tworzony jest model klasyfikacyjny, który pomaga określić czy nowy pacjent może być potencjalnie chory.

Skuteczność klasyfikacji zależna jest między innymi od ilości i jakości danych oraz od klasyfikatora (algorytmu klasyfikującego). Istnieje wiele różnych algorytmów klasyfikacji, takich jak: drzewo decyzyjne, naiwny klasyfikator Bayesa, sieć

neuronowa, maszyna wektorów nośnych oraz wiele innych. Algorytmy te osiągają różną skuteczność dla różnych danych. W celu poprawy skuteczności klasyfikacji można wykorzystać meta-metody, inaczej metauczenie (ang. *meta learning*). Celem metauczenia jest poprawa skuteczności klasyfikacji istniejących algorytmów. Najczęściej wykorzystuje się przewidywania różnych klasyfikatorów, które łączy się tworząc metadane. Następnie meta-klasyfikator (główny klasyfikator) lub wszystkie klasyfikatory poprzez głosowanie decydują o końcowej klasie nowych obserwacji.

Dane medyczne często zawierają dużą ilość osób zdrowych, a małą ilość osób chorych. Klasyfikując nowych pacjentów zależy nam na dużej skuteczności wykrywania osób potencjalnie chorych. Niestety, w przypadku dużego niezrównoważenia klas danych, klasyfikator często osiąga niską skuteczność klasyfikacji mniej liczebnej klasy. Aby poprawić wykrywalność mniejszej klasy stosuje się sztuczne równoważenie zbiorów lub zmodyfikowane algorytmy klasyfikacji.

Zakres i cel pracy

Celem pracy magisterskiej było zbadanie wpływu meta-metod (metauczenia) na poprawę jakości klasyfikacji danych oraz danych niezrównoważonych. Do badań wybrano algorytmy bagging, boosting oraz stacking. Wszystkie meta-metody miały zostać przetestowane z kilkoma podstawowymi algorytmami klasyfikacji. Klasyfikacja miała być przeprowadzona dla danych z różnych dziedzin, z rożną liczbą przykładów i atrybutów oraz z różnym rozkładem klas. Ważnym elementem badań była klasyfikacja zbiorów niezrównoważonych, z wyraźną dominacją liczebną jednej klasy. We wstępnym przetwarzaniu danych miały zostać użyte metody równoważące rozkład klas w zbiorze danych. Metody równoważenia klas danych, należało porównać i zbadać wpływ na proces klasyfikacji z meta-metodami. Ostatnim elementem niniejszej pracy była próba stworzenia uniwersalnego meta-klasyfikatora, uzyskującego wysoką jakość klasyfikacji dla większości zbiorów danych.

Zakres pracy stanowiło przetestowanie meta-metod. Gotowe algorytmy klasyfikacji miały pochodzić z biblioteki $scikit-learn$. Implementacja algorytmów klasyfikacji nie wchodziła w zakres pracy. Miały zostać napisane testy w języku *python* badające, oceniające i porównujące meta-metody oraz wybrane klasyfikatory. Testy powinny zwracać otrzymane wyniki w postaci plików pdf. Ostatnim elementem było stworzenie meta-klasyfikatora.

Rozdział pierwszy i drugi stanowią wprowadzenie teoretyczne do problemu klasyfikacji oraz klasyfikacji danych niezrównoważonych. W rozdziałach tych opisana jest klasyfikacja, użyte algorytmy klasyfikacji oraz meta-metody. Czytelnik może zaznajomić się także z informacją w jaki sposób i z jakimi miarami ocenia się klasyfikację danych. W rozdziale poświęconym klasyfikacji danych niezrównoważonych, znajduje się geneza problemu oraz opis dostępnych i możliwych rozwiązań poprawy klasyfikacji takich danych.

Rozdział trzeci poświęcony jest opisowi platformy do badań oraz analizie użytych danych do klasyfikacji. Zostały także przetestowane i wybrane wiarygodne sposoby oceniania klasyfikatora oraz sposoby równoważenia zbiorów danych.

W rozdziale czwartym opisano przeprowadzone badania na meta-metodach. Przeprowadzono testy z różnymi algorytmami oraz ustawieniami klasyfikatorów. Część rozdziału przeznaczona została na poprawę klasyfikacji danych mniejszościowych. Przetestowano i porównano wybrane metody równoważenia klas w zbiorach razem z meta-metodami.

W piątym rozdziale zaprezentowano zmodyfikowany pomysł głosowania, w postaci klasyfikatora eksperckiego. Przedstawiono także projekt meta-klasyfikatora. Podobnie jak poprzednio, oba klasyfikatory zostały przetestowane z użyciem prawdziwych zbiorów danych.

W ostatnim rozdziale opisano uzyskane wyniki oraz podsumowanie całej pracy.

Rozdział 1

Wstęp teoretyczny

1.1 Klasyfikacja danych

Klasyfikacja jest to proces przyporządkowania danych do jednej z predefiniowanych klas na podstawie atrybutów tych danych. Algorytm klasyfikacji na podstawie analizy danych trenujących, zawierających atrybuty oraz klasę, tworzy model klasyfikacyjny. Stworzony model klasyfikacyjny wykorzystywany jest do predykcji klasy (kategorii) nowych danych bez określonej klasy. Celem algorytmu budującego model, jest odnalezienie wzorców, w jaki sposób atrybuty obiektu wpływają na przynależność do danej klasy, tak, aby wiedza na temat analizowanych danych była możliwie ogólna oraz niezależna od próby.

Klasyfikacja danych jest procesem dwuetapowym:

- budowa modelu – proces ten polega na analizie obiektów z przyporządkowaną klasą oraz na budowie modelu opisującego predefiniowany zbiór klas danych,
- właściwa klasyfikacja – otrzymany model stosuje się do przydzielania klasy nowym obiektom.

Budowa modelu jest także procesem dwu-etapowym. Dzieli się ona na:

- uczenie – klasyfikator budowany jest w oparciu o dane treningowe,
- ocena jakości klasyfikacji – jakość klasyfikacji badana jest w oparciu o dane testowe.

W zależności od liczebności klas w zbiorze danych można wyróżnić:

- klasyfikację binarną – klasyfikator decyduje o przypisaniu obiektu do jednej z dwóch klas (np. czy człowiek jest zdrowy lub nie),
- klasyfikację wieloklasową – obiektowi przypisuje się jedną z wielu predefiniowanych klas.

Do reprezentacji danych uczących, testowych oraz do klasyfikacji najczęściej stosuje się system informacyjny.

Zachmurzenie	Temp.	Temp. wody	Opady	Wiatr	Pływać	h(x)
słonecznie	32	25	brak	słaby	tak	tak
słonecznie	31	26	brak	umiarkowany	tak	nie
pochmurnie	22	15	brak	b. mocny	nie	nie
pochmurnie	20	18	brak	słaby	tak	tak
całkowite zachmurzenie	12	6	brak	umiarkowany	nie	nie
całkowite zachmurzenie	10	8	duże	słaby	nie	nie
pochmurnie	21	10	brak	mocny	nie	tak
słonecznie	25	17	brak	umiarkowany	tak	nie
pochmurnie	23	17	przelotne	umiarkowany	nie	tak

Tablica 1.1: Przykład danych treningowych składających się z 5 atrybutów oraz klasy decyzyjnej. W ostatniej kolumnie znajduje się wynik klasyfikacji. W pięciu przypadkach, klasyfikator poprawnie wskazał klasę.

1.2 Wybrane algorytmy klasyfikacji danych

1.2.1 Drzewo decyzyjne

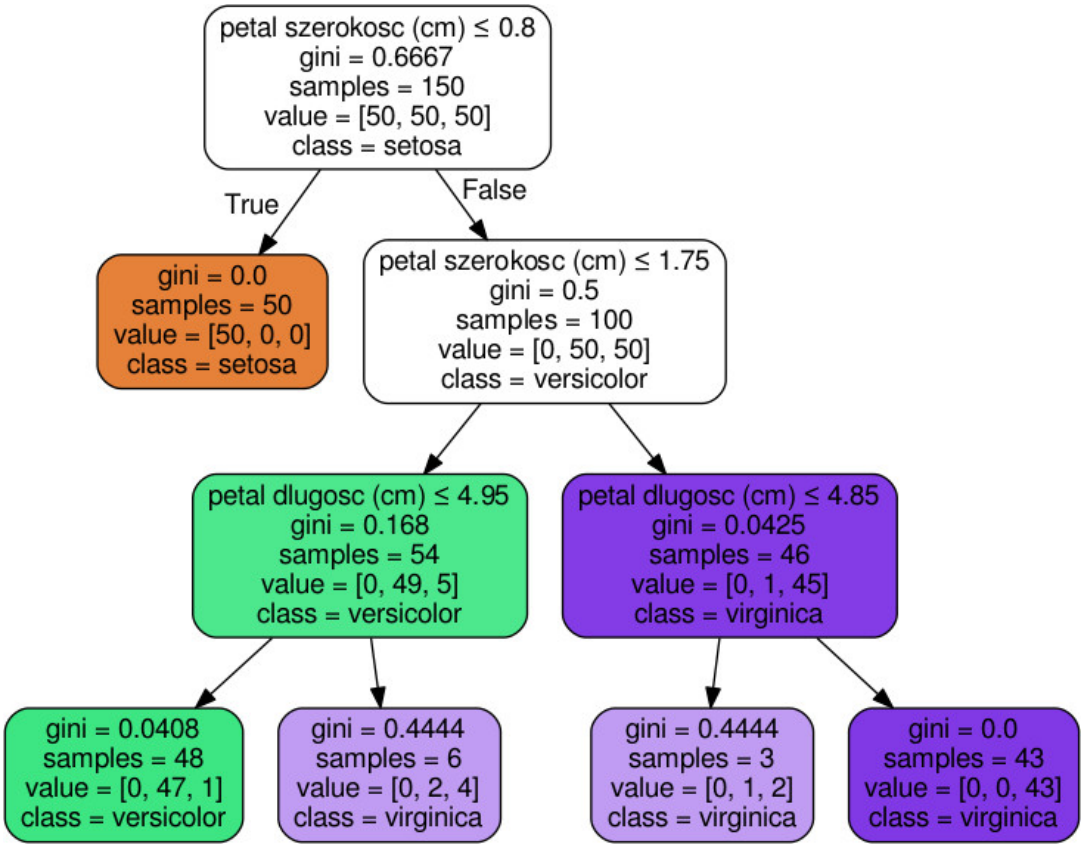
Drzewo decyzyjne jest bardzo często wykorzystywane jako klasyfikator danych. Celem jest stworzenie modelu, który na podstawie danych wejściowych przewidzi poprawnie klasę. Drzewo jest acyklicznym spójnym grafem skierowanym. Korzeń (węzeł na poziomie 0) zawiera w sobie cały zbiór uczący. W każdym węźle przeprowadza się test na wartościach atrybutu, który dzieli zbiór na podzbiory. Z węzła wychodzi tyle gałęzi ile możliwych jest wyników testu z tego węzła. Pod każdym węzłem znajduje się kryterium podziału dokonywanego w danym węźle, które jest jednakowe dla wszystkich elementów zbioru. Ostatnim elementem drzewa decyzyjnego są liście, które zawierają etykiety, czyli przydział klasowy elementów z tego podzbioru. Drzewo buduje się w sposób rekurencyjny od korzenia do liścia z wykorzystaniem metody „dziel i zwyciężaj".

Proces budowy drzewa:

1. Stwórz korzeń zawierający cały zbiór uczący.
2. Jeśli wszystkie przykłady należą do tej samej klasy decyzyjnej, to węzeł staje się liściem z etykietą klasy.
3. Jeżeli nie, oblicz kryterium podziału wykorzystując np. entropię, które najlepiej dzieli zbiór treningowy.
4. Dla każdego testu stwórz gałąź i podziel odpowiednio podzbiory do nowych węzłów.
5. Wywołaj rekurencyjnie algorytm dla nowych węzłów.
6. Algorytm kończy się dla kryterium stopu.

Stosuje się różne kryterium stopu:

- wszystkie przykłady należą do tej samej klasy,
- brak możliwości dalszego podziału,
- zbiór pusty,
- osiągnięto zakładany cel, np.: maksymalna głębokość drzewa, maksymalna czystość klas w liściu, minimalny przyrost informacji po podziale.



Rysunek 1.1: Drzewo decyzyjne z maksymalną głębokością równą 3. Drzewo zbudowane dla danych *iris-data*, zawiera trzy klasy (gatunki) setosa, versicolor i virginica oraz cztery atrybuty długość i szerokość kwiatu peta i sepal. Podział został dokonany ze współczynnikiem gini.

Jako kryterium podziału można zastosować np. wskaźnik Giniego lub entropię. Często zdarza się, że zbudowane drzewa są zbyt duże i tworzy się nadmierne dopasowanie do danych. Wówczas należy ograniczyć wysokość drzewa. Istnieje kilka algorytmów drzew decyzyjnych takich jak: ID3, C4.5, CART, CHAID. Przykładowe drzewo decyzyjne znajduje się na rysunku 1.1.

1.2.2 Naiwny klasyfikator bayesowski

Naiwny klasyfikator bayesowski opiera się na twierdzeniu o prawdopodobieństwie warunkowym stworzonym przez Thomasa Bayesa. Jest to klasyfikator probabilistyczny, zwracający prawdopodobieństwo przynależności przykładu do danej kategorii. Liczba kategorii musi być skończona i zdefiniowana a priori. Zasada działania klasyfikator opiera się na twierdzeniu:

$$P(C_i|x) = \frac{P(x|C_i)P(C_i)}{P(x)}$$

gdzie:

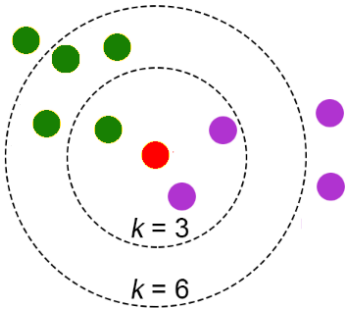
- x - przykład, dla którego nieznana jest klasa, jest to n-wymiarowy wektor, a n oznacza liczbę atrybutów,
- $P(C_i|x)$ - prawdopodobieństwo a posteriori, że przykład x należy do klasy C_i,
- $P(x|C_i)$ - prawdopodobieństwo a priori, że przykład x należy do klasy C_i,
- $P(C_i)$ - prawdopodobieństwo, że dowolny przykład należy do klasy C_i
- $P(x)$ to prawdopodobieństwo a priori wystąpienia przykładu x.

Działanie naiwnego klasyfikatora bayesowskiego oparte jest na założeniu, że atrybuty wewnątrz klasy są od siebie niezależne. Bardzo często to założenie nie jest spełnione, co rzutuje na wyniki osiągane przez ten klasyfikator oraz pokazuje genezę pochodzenia nazwy klasyfikatora.

1.2.3 Klasyfikator k najbliższych sąsiadów (kNN)

Klasyfikator k najbliższych sąsiadów lub klasyfikator k-NN, (ang. *k nearest neighbours*) należy do grupy algorytmów leniwych (ang. *lazy learners*), które nie wymagają uczenia, a całe obliczenia wykonywane są w momencie pojawienia się wzorca testowego, czyli podczas klasyfikacji lub testowania.

Klasyfikacja nowego obiektu x odbywa się poprzez znalezienie k najbliższych sąsiadów w danych uczących X i nadaniu mu nowej klasy poprzez głosowanie większościowe sąsiadów. Odległość pomiędzy sąsiadami można obliczyć np. takimi miarami jak: euklidesowa, Manhattan, Czebyszewa lub Minkowskiego.



Rysunek 1.2: Przykład klasyfikatora kNN. Klasyfikator ma przewidzieć klasę dla czerwonego punktu. Dla k=3 wyszukuje trzech najbliższych sąsiadów, w tym przypadku nadałby klasę fioletową, natomiast dla k=6 wyszukuje sześciu najbliższych sąsiadów i punktowi nadałby klasę zieloną.

Dobranie idealnej wartości parametru k ma bardzo duże znaczenie w jakości klasyfikacji. Dobrze dobrane k powinno być na tyle duże, aby minimalizować prawdopodobieństwo błędnych klasyfikacji, ale jednocześnie małe, aby k najbliższych sąsiadów było rzeczywiście bliskimi sąsiadami testowanej obserwacji [2].

1.2.4 Las losowy

Las losowy (ang. *random forest*) lub losowe drzewa decyzyjne [4] to klasyfikator złożony z drzew decyzyjnych. Algorytm trenujący działa według następującego schematu:

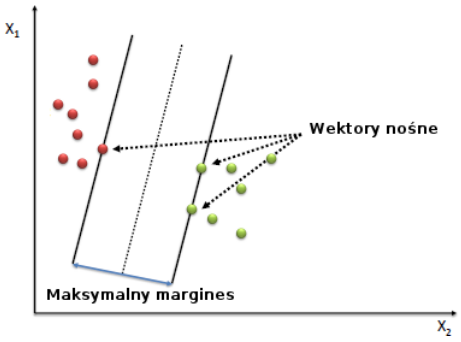
1. Wylosuj metodą boostrap (losowanie ze zwracaniem) n elementów ze zbioru danych.
2. Zbuduj drzewo decyzyjne, w każdym węźle:
   - wylosuj d atrybutów,
   - dla wylosowanych atrybutów, dokonaj najlepszego podziału (np. używając entropii, lub współczynnika Giniego).
3. Powtórz punkt 1 i 2 k razy (gdzie k, to liczba drzew).

Każde drzewo klasyfikuje przykład, a ostateczna klasa nadawana jest poprzez głosowanie większościowe.

Zazwyczaj, im większa liczba drzew k tym lepsze wyniki można otrzymać. Zmieniając liczbę losowanych przykładów n do zbioru uczącego, można kontrolować wariancję błędu. Im n większe, tym las losowy ma większą tendencję do nadmiernego dopasowania. Zmniejszając liczbę losowanych obserwacji można zmniejszyć szansę na przeuczenie i podnieść jakość klasyfikacji. Zazwyczaj n równa się wielkości zbioru danych. Liczbę losowanych atrybutów d, zwykle przyjmuje się na poziomie $d = \sqrt{m}$, gdzie m to liczba wszystkich atrybutów.

1.2.5 Maszyna wektorów nośnych

Maszyna wektorów nośnych, SVM (ang. *support vector machine*) jest to nieprobabilistyczny, binarny, liniowy klasyfikator. Zbudowany model SVM określa, do której z dwóch klas należą dane wejściowe. Konstrukcja klasyfikatora opiera się na znalezieniu hiperpłaszczyzny w przestrzeni wielowymiarowej oddzielającej dwie klasy zbioru uczącego. Zadaniem algorytmu jest wybór tej z możliwie największym marginesem, tak aby odległość najbliższych punktów po obu stronach była największa. Oddzielająca hiperpłaszczyzna, opisana jest z wykorzystaniem wektora normalnego. Wektor ten jest liniową kombinacją najbliższych do hiperpłaszczyzny punktów. Jakość klasyfikacji zależy od szerokości granicy rozdzielającej klasy, im jest większa, tym błąd uogólnienia powinien być mniejszy. Modele z małym marginesem mogą wykazywać nadmierne dopasowanie. Klasyfikacja właściwa nowych przypadków, polega na określeniu po której stronie marginesu znajduje się nowy punkt i na tej podstawie wyznaczana jest klasa.



Rysunek 1.3: Przykład klasyfikatora SVM z maksymalnym marginesem oddzielającym dwie klasy.

Istnieje modyfikacja podstawowego algorytmu, dla przypadku, gdy nie istnieje hiperpłaszczyzna oddzielająca dwie klasy. Jest to maszyna wektorów nośnych o miękkim marginesie. Bierze ona pod uwagę możliwość występowania zaszumionych danych. Klasyfikator w procesie uczenia, szuka możliwie najlepszej hiperpłaszczyzny oddzielającej obie klasy. Stopień błędnej klasyfikacji mierzony jest poprzez zmienną rozluźniającą (ang. *slack variable*). Zmienną tą można traktować jako karę, dla danych znajdujących się po złej stronie granicy. Można ją kontrolować poprzez parametr C, który decyduje o szerokości rozdzielającego marginesu. Duża wartość parametru C oznacza wysokie kary za błędną klasyfikację.

Jeżeli problem jest nieseparowalny liniowo, można zastosować tzw. trik jądra, który polega na zastąpieniu liniowego jądra, nieliniowym. Pozwala to na stworzenie klasyfikatora nieliniowego. Najczęściej stosuje się następujące jądra:

- wielomianowe (ang. *polynomial*),
- potencjalnych funkcji bazowych RBF (ang. *radial basis function*),
- sigmoidalne.

1.2.6 Zespół klasyfikatorów

W celu poprawy jakości klasyfikacji, można zastosować zespół klasyfikatorów w celu przypisania kategorii nowemu przykładowi. Zespół taki może składać się z kilku klasyfikatorów o takich samych algorytmach lub różnych. Jeżeli są to takie same klasyfikatory, to każdy model uczony jest na innych danych. Dla każdego modelu, dane trenujące mogą być wyznaczane poprzez losowanie lub losowanie ze zwracaniem. Zwykle zbiór trenujący przyjmuje nie więcej niż 2/3 wszystkich elementów. Możliwe jest także podzielenie zbioru trenującego na $k + 1$ (gdzie k to liczba klasyfikatorów) części i przekazanie każdemu klasyfikatorowi różnych k części jako zbiór trenujący. Innym sposobem jest modyfikacja przestrzeni atrybutów. Każdy z klasyfikatorów otrzymuje zbiór danych zawierający inne atrybuty. Jeżeli są to różne algorytmy, to wszystkie modele można uczyć na takich samych danych. Każdy klasyfikator otrzymuje nowy przykład i nadaje mu kategorię. Ostateczna kategoria wyznaczana jest poprzez:

- głosowanie większościowe (ang. *majority voting*) wybierana jest klasa najczęściej wskazywana przez modele,
- głosowanie ważone (ang. *weighted voting*) każdy klasyfikator ma przypisaną wagę lub zamiast predykcji klasy, zwraca prawdopodobieństwo z jakim przewiduje daną klasę. Docelową klasą jest ta z największym prawdopodobieństwem.

1.3 Meta-metody

1.3.1 Bagging

Metoda bagging [3], znana także jako bootstrap aggregating to zespół klasyfikatorów, meta-algorytm pozwalający zmniejszyć wariancję oraz uniknąć nadmiernego dopasowania. Metoda ta została zaproponowana w 1994 przez Leo Breiman'a w celu podniesienia jakości klasyfikacji poprzez połączenie wielu klasyfikatorów tego samego typu, uczących się na losowym zbiorze danych. Jest to połączenie wielu klasyfikatorów opartych o ten sam algorytm, ale trenowanych na różnych zbiorach danych. Zbiór uczący dla każdego klasyfikatora tworzy się poprzez losowanie ze zwracaniem z głównego zbioru danych. Losowanie wykonuje się z rozkładem jednostajnym. Nowo tworzony zbiór może być mniejszy lub takiej samej wielkości. Jeżeli zbiór danych jest duży i wygenerowany zbiór ma taką samą wielkość, można spodziewać się $1-\frac{1}{e}$ (około 63,2%) unikalnych próbek. Takie losowanie nosi nazwę próby bootstrap. Zazwyczaj łączy się wyniki wielu modeli tego samego typu. Mając zbiór uczący D o rozmiarze n, algorytm wygląda następująco:

1. Wygeneruj m nowych zbiorów treningowych D_i, o rozmiarze n, poprzez losowanie ze zwracaniem ze zbioru D,
2. Trenuj m klasyfikatorów w oparciu o wygenerowane zbiory treningowe D_i.

Nowa próbka klasyfikowana jest przez wszystkie modele, a ostateczna klasa nadawana jest poprzez głosowanie większościowe. W metodzie bagging bardzo ważny jest dobór odpowiedniego klasyfikatora oraz rozważenie czy ta metoda może podnieść dokładność klasyfikacji. Jeżeli klasyfikator jest niestabilny (np. mała zmiana w danych treningowych powoduje zmianę dokładności klasyfikatora) metoda bagging może znacząco podnieść jakość i dokładność klasyfikacji. Jeżeli jednak klasyfikator jest stabilny, stosowanie metody bagging może doprowadzić do zmniejszenia jakości klasyfikacji z powodu zmniejszenia ilości danych treningowych.

1.3.2 Boosting

W 1988 i 1989 roku Micheal Kearns oraz Leslie Valiant postawili pytanie czy można stworzyć zbiór słabych klasyfikatorów, w celu stworzenia jednego silnego. Słaby klasyfikator to taki, który osiąga tylko minimalnie lepsze wyniki od losowania (zgadywania), natomiast dobry klasyfikator osiąga wyniki mocno skorelowane z rzeczywistą klasyfikacją. W 1996 roku Robert Schapire oraz Yoav Freund zaprezentowali algorytm AdaBoost [6]. Istnieje dużo różnych algorytmów boostingu. Metoda boosting [7] to zbiór klasyfikatorów, meta-algorytm, w którym w odróżnieniu od baggingu, słabe klasyfikatory budowane są sekwencyjnie. Klasyfikatory budowane są w oparciu o ten sam algorytm klasyfikacyjny, ale trenowane są na zbiorach z różnymi wagami przykładów. Większość algorytmów boostingu działa według następującego schematu:

1. Każdemu przykładowi ze zbioru początkowego przypisywana jest taka sama waga początkowa, zazwyczaj $\frac{1}{n}$.
2. Budowany jest nowy klasyfikator m w oparciu o zbiór z wagami.
3. Zbudowany klasyfikator dołączany jest do klasyfikatorów M z wagą odpowiadającą jego dokładności.
4. W zbiorze trenującym następuje aktualizacja wag. Przykładom poprawnie sklasyfikowanym zmniejsza się wagę, natomiast błędnie sklasyfikowanym zwiększa się.
5. Punkt 2-4 powtarzany jest do momentu osiągnięcia liczby estymatorów m lub do osiągnięcia zakładanego błędu.

Takie podejście nazywane jest boosting by weighting, w procesie uczenia bierze udział cały zbiór. Istnieje także boosting by sampling, w którym zbiór jest ograniczony do n elementów, a zamiast wag, używa się prawdopodobieństwa. Zwiększając prawdopodobieństwo przykładów źle sklasyfikowanych, zwiększa się szansę na wylosowanie, a w konsekwencji na poprawną klasyfikację przez model. W klasyfikacji końcowej udział biorą wszystkie klasyfikatory, a ostateczna klasa nadawana jest poprzez głosowanie ważone.

Algorytm boosting skupia się na przykładach błędnie klasyfikowanych. Pozwala znacząco poprawić jakość klasyfikacji w przypadku użycia słabych klasyfikatorów (skuteczność klasyfikacji niedużo powyżej 50%). W przypadku klasyfikatorów osiągających lepsze wyniki, nie obserwuje się znaczącego przyrostu skuteczności. Boosting może wykazywać nadmierne dopasowanie do przykładów wielokrotnie źle klasyfikowanych.

W 2008 roku Phillip Long (Google) oraz Rocco A. Serveido (Uniwersytet Columbia), podczas 25. Międzynarodowej Konferencji poświęconej systemom uczącym, opublikowali artykuł [11], w którym zasugerowali, że większość stosowanych współcześnie algorytmów opartych o boosting jest wadliwa. Stwierdzili, że algorytmy AdaBoost, LogiBoost mogą wykazywać słabą odporność na losowy szum klasyfikacyjny, a zastosowanie ich w realnym świecie jest wielce wątpliwe. W artykule przedstawiony został przykład, w którym stwierdzono, że jeżeli część danych treningowych będzie miała źle oznaczone klasy, to algorytm nie będzie mógł poprawnie sklasyfikować tych danych. Skutkiem czego stworzenie modelu z dokładnością większą niż $\frac{1}{2}$ będzie niemożliwe.

Istnieje dużo algorytmów korzystających z metody boosting. Najpopularniejsze z nich to AdaBoost, LPBoost, TotalBoost, BrownBoost, MadaBoost, LogitBoost.

1.3.3 Stacking

Metoda stacking (kontaminacja modeli) [18] to połączenie kilku lub kilkunastu klasyfikatorów o różnych algorytmach. W pierwszym etapie, klasyfikatory trenowane są na tych samych danych. Następnie zbiór treningowy klasyfikowany jest przez te modele, a przewidziane klasy tworzą nowy zbiór uczący dla meta-klasyfikatora. Zbiór treningowy można także utworzyć z połączenia danych wejściowych z predykowanymi klasami. Zamiast klas, jako dodatkowe wejście, można użyć prawdopodobieństwa danej klasy, o ile wszystkie klasyfikatory wspierają tą funkcję. Meta-klasyfikatorem może być każdy algorytm. Bardzo często do tego celu stosuje się regresję logistyczną lub sieć neuronową.

1.4 Wstępne przetwarzanie danych

1.4.1 Brakujące wartości atrybutów

Często występującym problemem jest niekompletność baz danych, brak kilku wartości różnych atrybutów. Brakujące wartości mogą być wynikiem błędu człowieka, aplikacji, programu pomiarowego, nie podania danych lub z innego powodu. Zazwyczaj brakujące dane oznaczone są pustymi polami, pytajnikiem lub w inny opisany sposób. Istnieje kilka sposobów na rozwiązanie tego problemu.

Usunięcie niekompletnych obserwacji

Najprostszym sposobem jest usunięcie wierszy lub kolumn, w których brakuje wartości. Przed usunięciem należy przeanalizować dane, sprawdzić, które usunięcie będzie najbardziej korzystne (usunie najmniej danych). Może zaistnieć sytuacja, że zamiast usuwać dużą liczbę przykładów (wiersze), bardziej opłaca usunąć się atrybut (kolumnę), który ma dużo pustych komórek. Opisana metoda może również mieć niepożądane konsekwencje. Usuwając niektóre obserwacje lub atrybuty, pozbywa się części informacji. Skutkiem tego zabiegu model predykcyjny może działać słabiej. Konsekwencją stosowania tego sposobu jest brak możliwości predykcji niekompletnych przykładów.

Imputacja danych

Innym pomysłem na rozwiązanie tego problemu jest imputacja danych. Brakujące dane można obliczyć lub wyznaczyć różnymi technikami na podstawie wartości pozostałych obserwacji. Jeżeli atrybut zawiera wartości ciągłe, brakujące elementy można zastąpić wartością średnią lub medianą całej kolumny. W przypadku wartości dyskretnych można uzupełnić je wartością występującą najczęściej. Stosując takie rozwiązanie można wprowadzić szum do danych. Rozwiązaniem dającym lepsze rezultaty, może być zastosowanie klasyfikatora lub regresji w celu imputacji danych.

1.4.2 Transformacja danych

Drzewo decyzyjne lub las losowy są jednymi z nielicznych algorytmów klasyfikacji, które nie wymagają skalowania danych. Atrybuty mogą mieć różne skale, jednostki i przedziały zmienności. Może to powodować dominację niektórych atrybutów nad innymi (np. w klasyfikatorze k-NN, podczas mierzenia odległości euklidesowej), a w konsekwencji do zafałszowania klasyfikacji. Ten problem można rozwiązać na kilka sposobów.

Skalowanie

Skalowanie polega na proporcjonalnym przekształceniu wartości atrybutów do nowego przedziału. Zazwyczaj jest to przedział [0,1]. Do przekształcenia wykorzystuje się następujący wzór:

$$x_i = p_{start} + (p_{stop} - p_{start})\frac{x_i - x_{min}}{x_{max} - x_{min}}$$

gdzie:

- x_i to nowa wartość atrybutu,
- x_{max} i x_{min} to wartość maksymalna i minimalna atrybutu,
- p_{start} i p_{stop} to granice przedziału docelowego.

Standaryzacja

Lepszym rozwiązaniem może być standaryzacja, z powodu uwarunkowania wielu algorytmów, które rozpoczynają klasyfikację od wag równych 0 lub bliskich 0. Wykorzystując standaryzację, dane są podobne do standardowego rozkładu normalnego, a średnie wartości atrybutów ustawiane są w zerze, co pozwala na łatwiejsze uczenie się wag. Zastosowanie standaryzacji czyni klasyfikator bardziej odpornym na przykłady poboczne (tzw. outliers), w przeciwieństwie do skalowania. Standaryzację wyraża się wzorem:

$$x_i^{STD} = \frac{x_i - \mu(x)}{\sigma(x)}$$

gdzie:

- x_i^{STD} to nowa wartość atrybutu,
- x_i to wartość początkowa,
- $\mu(x)$ to wartość średnia,
- $\sigma(x)$ to odchylenie standardowe.

1.4.3 Dane kategoryczne

Bardzo często zdarza się, że bazy danych oprócz danych numerycznych, zawierają także dane kategoryczne, zapisane w postaci łańcucha znaków. Wyróżnia się dane kategoryczne nominalne i porządkowe. Dane kategoryczne nominalne zawierają cechy wzajemnie się wykluczające. Przykładem może być płeć człowieka, kolor skóry, oczu, kolor koszulki. Takich danych nie można posortować ani też porównać. Natomiast dane kategoryczne porządkowe zawierają informacje jednoznaczne identyfikujące, które można uporządkować (np. rozmiar koszulki).

Mapowanie danych kategorycznych

Aby mieć pewność, że algorytm klasyfikacji odpowiednio zinterpretuje dane kategoryczne, należy je przekonwertować do wartości numerycznych. W poniższym przykładzie (tabela 1.2) kategoria płeć, kraj są przykładem danych kategorycznych nominalnych, a wzrost jest atrybutem porządkowym. Należy pamiętać, aby w przypadku danych porządkowych, odpowiednio odwzorować zależności. W procesie zmieniania wystarczy wykonać odpowiednie mapowanie. Przykładowe mapowanie przedstawiono poniżej.
Atrybut płeć:

- mężczyzna -> 0
- kobieta ->1

Atrybut kraj:

- Polska -> 0
- Niemcy ->1

Atrybut wzrost:

- niski -> 0
- średni -> 1
- wysoki -> 2

Atrybut klasa:

- klasa1 -> 0
- klasa2 -> 1

ID	Płeć	Kraj	Wzrost	Wiek	Klasa
1	Mężczyzna	Polska	Niski	25	klasa 1
2	Mężczyzna	Polska	Wysoki	52	klasa 2
3	Kobieta	Polska	Średni	19	klasa 2
4	Kobieta	Niemcy	Niski	37	klasa 1

Tablica 1.2: Przykład danych z kategorycznymi atrybutami.

Dane nominalne

Dane kategoryczne nominalne nie mogą zostać w takiej postaci. Dane te nie posiadają określonego porządku, ani nie można ich porównać. Zostawiając je w takiej formie algorytm mógłby porównać kobietę z mężczyzną. Wynik klasyfikacji mógłby być prawidłowy, ale niekoniecznie najlepszy.

Dla takich danych, należy stworzyć dodatkowe atrybuty o wartościach binarnych (nazwa tej metody w języku angielskim to *one hot encdoing*). W przypadku atrybutu płeć, należy stworzyć dwie kolumny „mężczyzna" i „kobieta". Poniżej w tabeli 1.3 przedstawiono poprawnie zakodowane dane.

ID	M	K	Pol	Nie	Wzrost	Wiek	Klasa
1	1	0	1	0	0	25	0
2	1	0	1	0	2	52	1
3	0	1	1	0	1	19	1
4	0	1	0	1	0	37	0

Tablica 1.3: Przykład danych z atrybutami kategorycznymi danymi po odpowiednim kodowaniu.

1.4.4 Redukcja liczby atrybutów

Jednym ze sposobów uniknięcia nadmiernego dopasowania klasyfikatora do danych lub zmniejszenia jego złożoności jest redukcja atrybutów. Zdarza się, że niektóre atrybuty mają znikomy wpływ na kategorię przykładu, a mogą stanowić szum klasyfikacyjny. Atrybuty mniej ważne usuwa się, a zostawia się tylko te z największą ilością informacji, mających największy wpływ na klasyfikację. Usunięcie bezużytecznych atrybutów może zwiększyć skuteczność klasyfikacji. Kolejną możliwością jest redukcja wymiarów, czyli poszukiwanie wzorców danych, tak aby kilka atrybutów zgrupować w jeden. Poniżej przedstawiono kilka różnych algorytmów.

Sekwencyjna selekcja postępująca oraz sekwencyjna selekcja wsteczna

Sekwencyjna selekcja postępująca (ang. *sequential forward selection* (SFS)), algorytm zaczyna działanie od pustego zbioru atrybutów. W kolejnych iteracjach dodaje atrybut, który maksymalizuje wybraną funkcję (najczęściej jest to dokładność klasyfikacji). Innymi słowy, dodaje te atrybuty, które najbardziej podnoszą jakość klasyfikacji. Algorytm działa do momentu osiągnięcia zakładanej liczby atrybutów lub do pogorszenia skuteczności klasyfikacji. Algorytm ten posiada wadę, gdyż raz dodana cecha nie może już zostać usunięta. Może zdarzyć się taka sytuacja, że dodany już atrybut powoduje obniżenie jakości po dodaniu nowego atrybutu. Rozwiązaniem tego problemu jest algorytm SFFS (ang. *sequential floating forward selection*), który po dodaniu nowej cechy usuwa inny atrybut jeśli zwiększy to wartość funkcji.

Sekwencyjna selekcja wsteczna (ang. *sequential backward selection* (SBS)), algorytm usuwa kolejne atrybuty powodujące najmniejszy spadek w jakości klasyfikatora (mające najmniejszy wpływ), do momentu osiągnięcia pożądanej liczby atrybutów. Istnieje także rozszerzenie tego algorytmu w postaci SFBS (ang. *sequential floating backward selection*)). W tym algorytmie, po usunięciu atrybutu, w kolejnym kroku sprawdza się, czy usunięte wcześniej już atrybuty po ponownym dodaniu nie podniosą wartości wybranej funkcji.

Przedstawione algorytmy są algorytmami zachłannymi i bardzo czasochłonnymi. W pesymistycznym przypadku, klasyfikator może być budowany $\frac{p(p+1)}{2}$ razy.

Analiza głównych składowych

Analiza głównych składowych PCA (ang. *principal component analysis*) jest to nienadzorowana liniowa transformacja, używana między innymi do redukcji wymiarów danych. Bardzo często stosuje się ją w statystyce oraz analizie danych. PCA identyfikuje wzorce w danych pomiędzy atrybutami. Algorytm szuka maksymalnych kierunków wariancji wielowymiarowych danych i następnie rzutuje je w nowej podprzestrzeni z mniejszą lub taką samą liczbą atrybutów.

1.5 Ocena poprawności klasyfikacji

1.5.1 Miary jakości klasyfikacji danych

Jakość klasyfikacji można ocenić na podstawie kilku współczynników. Do ich obliczenia wykorzystuje się macierz pomyłek (tabela 1.4). Tworzona jest ona w oparciu o wynik klasyfikacji. Dla klasyfikacji binarnej macierz składa się z dwóch kolumn oraz dwóch wierszy. W wierszach znajdują się poprawne klasy decyzyjne, natomiast w kolumnach przewidziane przez klasyfikator. Zaklasyfikowane obiekty, umieszcza się w odpowiedniej grupie.

		Klasa predykowana	
		pozytywna	negatywna
Klasa rzeczywista	pozytywna	prawdziwie pozytywna (TP)	fałszywie negatywna (FN)
	negatywna	fałszywie pozytywna (FP)	prawdziwie negatywna (TN)

Tablica 1.4: Macierz pomyłek.

Nazwy grup inspirowane były nazewnictwem medycznym. Dla dwóch klas w zbiorze danych wyróżnia się następujące grupy:

- prawdziwie pozytywna (ang. *true positive*), skrót TP: są to obiekty należące do klasy pozytywnej oraz zakwalifikowane przez klasyfikator jako pozytywne (trafienie, z ang. *hit*),
- fałszywie negatywna (ang. *false negative*), skrót FN: są to obiekty należące do klasy pozytywnej, ale zostały błędnie zakwalifikowane przez klasyfikator jako negatywne (błąd pominięcia, z ang. *miss*),
- fałszywie pozytywna (ang. *false positive*), skrót FP: są to obiekty należące do klasy negatywnej, błędnie uznane przez klasyfikator jako pozytywne (fałszywy alarm, ang. *false alarm*),
- prawdziwie negatywna (ang. *true negative*), skrót TN: są to obiekty należące klasy do negatywnej, i sklasyfikowane przez klasyfikator jako negatywne (poprawnie odrzucone, ang. *correct rejection*).

Ocenę jakości klasyfikacji przeprowadza się w oparciu o współczynniki wyliczane na podstawie macierzy pomyłek.

Podstawowym kryterium służącym do oceny klasyfikacji jest dokładność (ang. *accuracy*), jest to stosunek wszystkich poprawnie sklasyfikowanych przykładów klasy pozytywnej oraz negatywnej do wszystkich przykładów. Miara ta określa dokładność z jaką klasyfikator podaje poprawny wynik.

$$accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

Można wyróżnić także błąd klasyfikatora, obliczany na podstawie dokładności.

$$Error\ rate = 1 - accuracy$$

Trzecim wskaźnikiem oceny klasyfikacji jest TPR (ang. *true positive rate*), często określany jako czułość (ang. *sensitivity* lub *recall*). Jest to stosunek obiektów poprawnie sklasyfikowanych jako pozytywne z wszystkimi pozytywnymi przykładami. Wskaźnik ten pokazuje poprawność klasyfikowania obserwacji pozytywnych. Wykorzystując tę miarę w medycynie można określać skuteczność wykrywania osób chorych.

$$Sensitivity,\ Recall,\ TPR = \frac{TP}{TP + FN}$$

Kolejną miarą oceniającą klasyfikację jest TNR (ang. *true negative rate*), nazywana także specyficznością (ang. *specificity.*). Wskazuje ona efektywność klasyfikowania przykładów negatywnych. Jest to stosunek poprawnie przydzielonych przykładów negatywnych do wszystkich negatywnych obserwacji. Z jej pomocą, można ocenić celność klasyfikacji osób zdrowych.

$$Specificity,\ TNR = \frac{TN}{TN + FP}$$

Wyróżnia się także współczynnik FPR (ang. *false positive rate*), jest to iloraz przykładów fałszywie pozytywnych i sumy przykładów prawdziwie negatywnych i fałszywie negatywnych.

$$FPR = \frac{FP}{TN + FP}$$

Istotnym wskaźnikiem jest także precyzja (ang. *precision*). Określa ona jaka część przykładów uznanych za pozytywne przez klasyfikator została poprawnie oznaczona. Precyzja wyrażana jest jako stosunek prawdziwie pozytywnych przypadków do wszystkich przykładów uznanych za pozytywne. W medycynie, pokazuje procentowo ile osób uznanych za chorych, jest rzeczywiście chora.

$$precision = \frac{TP}{TP + FP}$$

Wskaźnik dokładności oraz error rate nie sprawdzają się w przypadku, gdy dane są niezrównoważone. Klasyfikator może osiągnąć wysoką dokładność np. 90% przy niskiej wykrywalności klasy mniejszościowej (baza *balancescale* zawiera 92% próbek klasy 0 oraz 8% klasy 1, większość klasyfikatorów osiąga dokładność na poziomie 92% przy zerowej wykrywalności klasy 1).

Dlatego oceniając klasyfikator pracujący na niezrównoważonych danych, należy obliczyć osobno współczynniki precyzji, czułości oraz specyficzności dla każdej kategorii danych. Jak wspomniano wcześniej, bardzo często polepszenie jakości klasyfikacji klasy mniejszościowej połączona jest z pogorszeniem rozpoznawalności klasy większościowej. Mając współczynnik czułości oraz specyficzności ciężko zdecydować,

który klasyfikator jest lepszy. Kubat i Matwin zaproponowali połączenie obu tych współczynników, w postaci średniej geometrycznej czułości oraz specyficzności [12].

$$G - mean = \sqrt{Sensitivity * Specificity}$$

Klasyfikator z wyższym G-mean, zapewnia lepszą rozpoznawalność obu klas, jednocześnie zachowując, aby dokładność w rozpoznawaniu obu klas była zbilansowana. Współczynnik ten jest niezależny od rozkładu klas w danych [9].

Ocenę klasyfikacji danych niezrównoważonych można dokonać także przy pomocy F-measure. Jest to średnia harmoniczna precyzji oraz czułości. Współczynnik F-measure można obliczyć dla obu klas. β wykorzystywana jest do określenia zależności pomiędzy precyzją oraz czułością.

$$F - measure = \frac{(1+\beta)^2 * precision * recall}{\beta^2 * precision + recall}$$

Zazwyczaj $\beta = 1$, wtedy:

$$F - measure = 2 * \frac{precision * recall}{precision + recall}$$

Podstawiając wzory pod precision oraz recall można otrzymać uproszczoną wersję miary F_1:

$$F_1 - measure = \frac{2 * TP}{2 * TP + FP + FN}$$

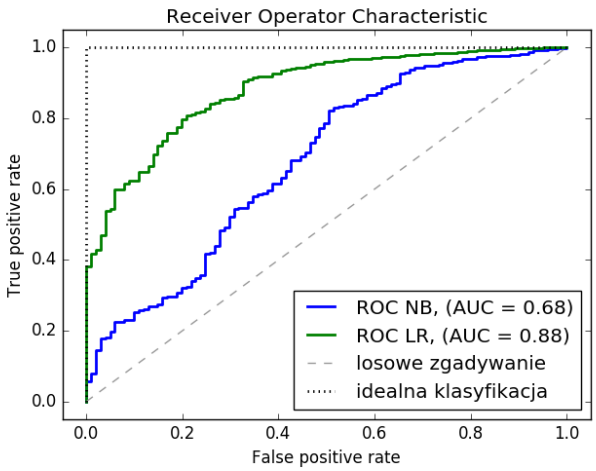
Krzywa ROC

Dla klasyfikatorów, które mogą zwracać prawdopodobieństwo klas, można zbudować wykres wartości TPR oraz FPR, które tworzą tzw. krzywą ROC (ang. *receiver operator characteristic*). Wykorzystując tą krzywą można porównać modele. Klasyfikator jest lepszy od drugiego, jeżeli jego krzywa jest powyżej drugiej krzywej. Jeżeli krzywa ROC przebiega poniżej przekątnej, to klasyfikator ma gorszą skuteczność niż losowe zgadywanie. Na wykresie 1.4 przedstawiono porównanie naiwnego klasyfikatora bayesowskiego oraz regresji logistycznej, która osiągnęła zdecydowanie lepszy wynik w klasyfikacji. Na wykresie zaznaczono krzywą ROC idealnego klasyfikatora, przechodzi ona przez punkty (1,0) oraz (1,1).

W celu porównania klasyfikatorów można obliczyć pole powierzchni poniżej krzywej ROC, jest to tzw. AUC (ang. *area under curve*). Idealny klasyfikator osiąga wartość AUC = 1, natomiast losowe zgadywanie AUC = 0.5. Im wartość AUC jest wyższa, tym model jest skuteczniejszy.

1.5.2 Nadmierne dopasowanie i wariancja klasyfikatora

Jeżeli model uzyskuje zdecydowanie lepsze wyniki na danych treningowych niż na danych testowych, świadczy to o nadmiernym dopasowaniu (ang. *overfitting*) klasyfikatora do danych uczących. Nadmierne dopasowanie występuje wtedy, gdy



Rysunek 1.4: Przykład krzywej ROC, dla naiwnego klasyfikatora bayesowskiego oraz dla regresji logistycznej dla danych *abalone*041629.

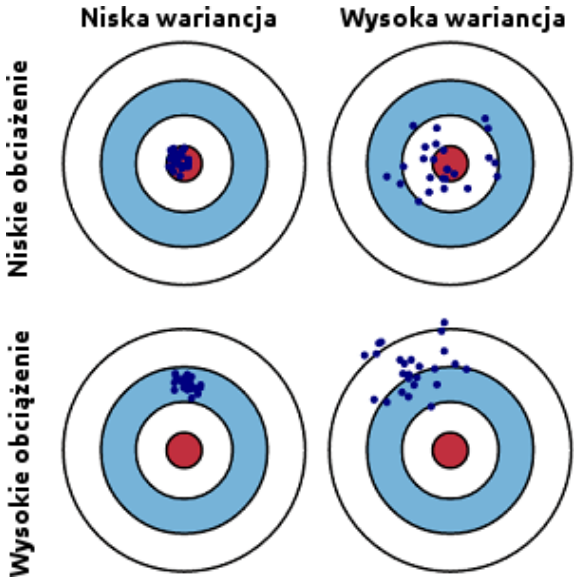
model zamiast dobrze generalizować prawdziwe dane, dopasowuje się za bardzo do danych treningowych. Powodem tego zjawiska jest zazwyczaj zbyt duża liczba parametrów (zbyt duża złożoność) modelu w stosunku do rozmiaru danych uczących. Klasyfikator może mieć wysoką skuteczność dla danych treningowych, jednak dla nowych danych będzie generował gorsze wyniki. Rozwiązaniem tego problemu może być:

- zebranie większej ilości danych uczących,
- wprowadzanie kary (np. regularyzacja L1 lub L2) za zbyt dużą złożoność modelu,
- wybranie prostszego algorytmu z mniejszą ilością parametrów,
- zmniejszenie wymiaru danych (usunięcie niektórych atrybutów).

Klasyfikator z nadmiernym dopasowaniem posiada wysoką wariancję. Wariancja mierzy zmienność przewidywań modelu dla określonego zbioru testowego z wykorzystaniem różnych zbiorów uczących (lub podzbiorów). Wysoka wariancja świadczy o dużej zmienności przewidywań klas dla różnych danych treningowych i jest to niepożądane zjawisko. Mała wariancja to mało zmian, w miarę stabilne przewidywanie klas dla tych samych próbek. Model z wysoką wariancją wrażliwy jest na losowość i szum danych.

Klasyfikator może być także niedouczony, niedopasowany dostatecznie (ang. *underfitting*), co oznacza, że model nie jest zbyt skomplikowany, aby znaleźć odpowiednie wzorce danych. W wyniku czego, osiąga słabe wyniki. Miarą towarzyszącą niedopasowaniu jest obciążenie (ang. *bias*). Klasyfikator testuje się na próbce danych, wielokrotnie budując model na tym samym zbiorze danych uczących. Następnie mierzy się jak bardzo przewidywania klasyfikatora różnią się od prawdziwych klas. Wysokie obciążenie oznacza niedotrenowanie oraz małą skuteczność klasyfikacji. Niskie obciążenie oznacza bardzo dobre przewidywania klas. Obciążenie to błąd systematyczny.

Idealny klasyfikator odnajduje wzorce w danych treningowych oraz dobrze je generalizuje, tak aby skutecznie klasyfikować nie widziane wcześniej próbki. Celem w klasyfikacji nadzorowanej, jest zbudowanie klasyfikatora z małym obciążeniem oraz niską wariancją. Niestety, zazwyczaj nie można osiągnąć obu celi. Modele parametryczne i liniowe często mają niskie obciążenie, a wysoką wariancję. Natomiast modele nieparametryczne i nieliniowe zazwyczaj mają duże obciążenie, ale niską wariancję. Obciążenie i wariancja są ze sobą połączone, zmniejszając jeden błąd, zwiększa się drugi. Tą zależność nazywa się przetargiem obciążenia i wariancji (ang. *bias-variance trade-off*).



Rysunek 1.5: Obciążenie i wariancja na przykładzie tarczy strzeleckiej. Klasyfikator poprawnie przewiduje klasy jeśli niebieskie kropki trafiają w środek tarczy.

Zmieniając parametry klasyfikatorów można wpływać na balans pomiędzy tymi błędami:

- w algorytmie kNN, który często ma małe obciążenie i wysoką wariancję, zwiększając liczbę sąsiadów k, można zmniejszyć wariancję, ale równocześnie zwiększyć błąd obciążenia,
- w klasyfikatorze SVM, zwiększając karę, parametr C można zwiększyć obciążenie, a zmniejszyć wariancję.

Aby wychwycić nadmierne dopasowanie lub niedotrenowanie należy zbiór danych podzielić na zbiór uczący oraz na zbiór testowy. W ten sposób można sprawdzić z jaką skutecznością klasyfikator będzie pracował na nowych danych. Istnieją różne metody pomiarowe, które zostały opisane w podrozdziale 1.5.3.

1.5.3 Metody pomiaru jakości klasyfikacji danych

W celu prawidłowej oceny możliwości klasyfikatora zbiór danych należy podzielić na zbiór uczący oraz na zbiór testowy. Taki podział stosuje się po to, aby spraw-

dzić czy zbudowany model działa dobrze dla danych, które nie były wykorzystane do jego budowy (nadmierne dopasowanie). Zwiększając zbiór testowy, zmniejsza się zbiór uczący i tym samym mniej informacji jest stosowanych do tworzenia modelu. Może to prowadzić do zaniżenia ogólnej oceny klasyfikatora. Jednocześnie mały zbiór testowy może być niewystarczający do prawidłowej oceny, która może być obarczona dużym błędem. Proces oceny należy rozpocząć od budowy modelu w oparciu o dane treningowe, a następnie wykonać klasyfikację testową wykorzystując do tego zbiór testowy. Następnie buduje się macierz pomyłek na podstawie sklasyfikowanych przypadków. Kolejnym krokiem jest obliczenie opisanych wcześniej współczynników (zob. podrozdział 1.5.1), w oparciu o tą macierz. Istnieją różne schematy postępowania, służące do oceny zbudowanego modelu.

Metoda z jednym zbiorem

Do budowy klasyfikatora wykorzystywany jest cały zbiór dostępnych danych. W procesie testowania bierze udział także cały zbiór danych. Metoda ta, nie jest zbyt wartościowa i prowadzi do zawyżenia jakości klasyfikatora. W przypadku nowych danych, taki model osiągnie gorsze wyniki niż wskazywałyby na to obliczone współczynniki.

Metoda z wydzielonym zbiorem testowym (ang. *the holdout method*)

W tej metodzie, zbiór danych dzielony jest w sposób losowy na dwie części. Użytkownik dobiera rozmiar zbioru uczącego (np. 80%) oraz zbioru testowego (np. 20%). Wadą takiego rozwiązania jest losowy rozkład klas w zbiorze testowym oraz zmniejszenie zbioru uczącego. Może to doprowadzić do sytuacji nadmiernego dopasowania (zawyżonych wyników) lub do niedoszacowania klasyfikatora. Ważne jest, aby nie używać ciągle tego samego zbioru testowego do wyboru modeli, ale dokonywać losowania przed każdą oceną.

Ulepszeniem tej metody może być równy rozkład klas w obu zbiorach, tak aby zostały zachowane proporcje z oryginalnego zbioru.

Sprawdzian krzyżowy z p przykładami (ang. *leave-p-out cross-validation*)

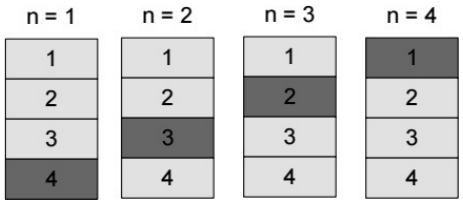
Sprawdzian krzyżowy z p przykładami wykorzystuje p obserwacji jako zbiór testowy, pozostałe elementy tworzą zbiór uczący. Cały proces jest powtarzany do momentu stworzenia i przetestowania wszystkich możliwych kombinacji p przykładów ze zbioru n. Ten rodzaj metody wymaga uczenia i testowania klasyfikatora $\binom{n}{p}$ razy, gdzie n to liczebność całego zbioru danych. W przypadku dużego zbioru danych oraz p>1, obliczenia mogą być czasochłonne, a nawet ze względu na dużą liczbę kombinacji, obliczenie ich może być niemożliwe.

Sprawdzian krzyżowy minus jeden element (ang. *leave-one-out cross-validation*)

Jest to specjalny przypadek sprawdzianu krzyżowego z p przykładami, dla p = 1. W tej metodzie zbiór testowy tworzy jeden element, pozostałe tworzą zbiór uczący. Testowanie klasyfikatora trwa do momentu użycia wszystkich obserwacji jako zbioru testowego. W przeciwieństwie do poprzedniej metody, ta jest wolna od czasochłonnych obliczeń, gdyż $\binom{n}{1}$=n, gdzie n to liczba wszystkich obserwacji. Zazwyczaj ta metoda wykorzystywana jest tylko do małych zbiorów danych.

Sprawdzian krzyżowy k-krotny (ang. *k-fold cross-validation*)

Zbiór danych jest losowo dzielony na k równych podzbiorów. Następnie każdy z podzbiorów w kolejnych k iteracjach staje się kolejno zbiorem testowym, pozostałe zbiory tworzą zbiór uczący, na podstawie którego buduje się model. Klasyfikacja i testowanie wykonywane są k-krotnie. Otrzymane wyniki łączy się i uśrednia w celu uzyskania jednego wyniku. Zaletą tej metody jest mały błąd estymacji oraz niższa wariancja błędu niż w przypadku metody minus jednego elementu. Zwykle stosuje się $k = 3..10$, dla których koszt czasowy jest umiarkowany. Sprawdzian krzyżowy często wykorzystywany jest do badania modeli, ze względu na możliwość przetestowania klasyfikatora na wszystkich danych.



Rysunek 1.6: Przykład sprawdzianu krzyżowego k-krotnego, $k = 4$.

Równomierny sprawdzian krzyżowy k-krotny (ang. *Stratified k-fold cross-validation*)

Jest to specjalny przypadek sprawdzianu krzyżowego k-krotnego. Podzbiory tworzone są z zachowaniem proporcji wszystkich klas. Każdy podzbiór powinien zawierać w przybliżeniu podobny procent obserwacji z każdej kategorii. Równomierny sprawdzian krzyżowy stosuje się zazwyczaj przy mniejszych zbiorach danych, zawierających klasy mniejszościowe, stanowiące mniej niż 10% ogółu danych.

Rozdział 2

Klasyfikacja danych niezrównoważonych

Większość istniejących algorytmów klasyfikacji, nastawiona jest na poprawną klasyfikację zbiorów o zrównoważonej liczebności wszystkich klas. Niestety, w rzeczywistych problemach, bardzo często zdarza się, że zbiory są mocno niezbilansowane. Istnieją dwie metody pozwalające zwiększyć skuteczność klasyfikacji danych mniejszościowych. W pierwszej metodzie modyfikuje się dane treningowe przed procesem uczenia. Dodaje się lub usuwa przykłady w celu zrównoważenia liczebnie obu klasy. W drugim podejściu wykorzystuje się zmodyfikowane algorytmy pod kątem niezrównoważonych zbiorów. W niniejszej pracy skupiono się wyłącznie na pierwszej metodzie, równoważeniu zbiorów.

2.1 Dane niezrównoważone

Dane są niezrównoważone (ang. *imbalanced data*) jeśli klasy decyzyjne nie są w przybliżeniu tak samo liczebne. Najmniejsza klasa nazywana jest klasą mniejszościową (ang. *minority class*), natomiast klasa dominująca, lub pozostałe połączone klasy (można połączyć pozostałe klasy w jedną, doprowadzając do klasyfikacji binarnej, one vs all), nazywana jest klasą większościową (ang. *majority class*). W praktyce klasa mniejszościowa zazwyczaj liczy około 10-20% wszystkich przykładów. Często zdarzają się jednak takie problemy, gdzie to zróżnicowanie jest większe np.:

- około 2% transakcji kartami kredytowymi w GOCARDLESS to oszustwa [20],
- dane medyczne zawierają bardzo dużą liczbę zdrowych pacjentów, a liczba chorych to najczęściej kilka procent lub mniej,
- około 1% rocznie twardych dysków ulega awarii.

W przytoczonych przykładach ważniejsza jest klasa mniejszościowa i wykrycie jej stanowi priorytet. Niezrównoważenie klas w zbiorze danych stanowi problem w fazie uczenia i znacząco obniża jakość klasyfikacji. Ze względu na częstość występowania

klasy dominującej, klasyfikator preferuje tą klasę, dążąc do optymalizacji i obniżenia błędu error rate (rozdział 1.5.1) nie biorąc pod uwagę rozłożenia klas w zbiorze. Klasyfikator może osiągnąć wysoką skuteczność klasyfikacji np. 95%, przy niskiej lub zerowej wykrywalności klasy mniejszościowej. Należy oczekiwać od klasyfikatora wysokiej skuteczności wykrywania klasy mniejszościowej, nawet kosztem pogorszenia rozpoznawania klasy większościowej. Przykłady z klasy zdominowanej można podzielić na cztery grupy. K. Napierała i J. Stefanowski wyróżnili przykłady[14]:

- safe - przykład bezpieczny, w jego sąsiedztwie zdecydowana większość obserwacji jest z tej samej klasy,
- borderline - graniczny, przykład niebezpieczny, w jego sąsiedztwie liczba przykładów z obu klas jest podobna
- outlier - poboczny, przykład niebezpieczny, w jego sąsiedztwie większość obserwacji jest z klasy przeciwnej, dominującej,
- rare - rzadki, przykład niebezpieczny, w jego sąsiedztwie występują tylko przykłady z klasy przeciwnej, większościowej.

2.2 Równoważenie rozkładu klas w zbiorze danych

W celu zrównoważenia rozkładu danych niezbilansowanych wprowadzono różne metody usuwania przykładów klasy dominującej lub tworzenia sztucznych obserwacji klasy mniejszościowej. Poniżej zostaną omówione metody, które zostały użyte podczas badań.

2.2.1 Metody undersampling

Jest to cała rodzina różnych metod, które usuwają przykłady z klasy większościowej. **Losowe usuwanie** (ang. *random undersampling*), jak sama nazwa wskazuje, usuwa losowo przykłady z klasy dominującej. Rozwiązanie to ma niestety wadę. Jeśli usunie się zbyt dużo przykładów danego przypadku, można pozbawić klasyfikator bardzo ważnej informacji.

Lepszym rozwiązaniem jest świadome usuwanie przykładów spełniających określone kryteria. Taką metodą jest **undersampling z „Tomek links”** [16]. Parę punktów Tomek link, definiuje się jako dwa punkty należące do różnych klas, z odległością równą $d(E_i, E_j)$, jeśli nie istnieje inny punkt E_l, taki, że $d(E_i, E_l) < d(E_i, E_j)$ lub $d(E_j, E_l) < d(E_i, E_j)$. Punkty tworzące Tomek link to szum lub punkt graniczny. Po znalezieniu takich punktów, usuwa się przykład z klasy dominującej. Usunięcie takiej obserwacji powoduje rozszerzenie granicy klasy mniejszościowej.

W metodzie ***Edited Nearest Neighbour*** (ENN) [17] analizowane są przykłady klasy większościowej. Usuwany jest każdy niewiarygodny przykład, jeżeli z trzech sąsiadów, przynajmniej dwóch ma inną klasę.

Modyfikacją metody ENN jest bardziej rygorystyczna reguła ***Neighbour Cleaning Rule*** (NCR) [13]. Usuwa ona zdecydowanie więcej przykładów z klasy większościowej. Usuwany jest każdy przykład dominującej klasy, jeżeli w jego sąsiedztwie znajdują się przynajmniej dwie obserwacje z klasy mniejszościowej. Dodatkowo, jeżeli w otoczeniu przykładu klasy zdominowanej znajdują się dwa przykłady z klasy dominującej, to te dwa przykłady również są usuwane.

2.2.2 Metody oversampling

Metody z rodziny oversampling generują nowe sztuczne obserwacje klasy mniejszościowej. Najprostszą metodą tworzenia nowych przykładów jest **losowe próbkowanie** (ang. *random oversampling*). Polega ona na kopiowaniu losowo wybranych przykładów z klasy mniejszościowej. Metoda ta, może doprowadzić do nadmiernego dopasowania, szczególnie w przypadku wielokrotnego skopiowania przykładów stanowiących szum.

Lepszym wyborem niż losowe tworzenie próbek może być metoda ***SMOTE*** (*Synthetic Minority Over-sampling Technique*) [5], która generuje nowe sztuczne obserwacje. W metodzie tej, analizowanych jest k najbliższych sąsiadów obserwacji z klasy mniejszościowej. Następnie generowane są nowe sztuczne przykłady, na losowo wybranych punktach z odcinków łączących analizowany przykład z sąsiadami. Liczbę wygenerowanych próbek można zdefiniować, w zależności od potrzebnej liczby nowych obserwacji. SMOTE nie analizuje przykładów z drugiej klasy, co może prowadzić do większego wymieszania obu klas, a w konsekwencji do pogorszenia skuteczności klasyfikacji.

Rozwinięciem metody SMOTE jest algorytm ***ADASYN*** (*Adaptive Synthetic Sampling Approach for Imbalanced Learning*) [8]. Metoda ADASYN podobnie jak SMOTE, generuje nowe sztuczne obserwacje klasy mniejszościowej, jednak skupia się bardziej na przykładach trudniejszych w klasyfikacji. Algorytm SMOTE generuje taką samą liczbę nowych próbek dla każdego przykładu klasy zdominowanej. ADASYN wprowadza różne wagi dla obserwacji klasy mniejszościowej, dzięki czemu można skuteczniej klasyfikować trudniejsze przykłady.

2.2.3 Metody hybrydowe

W celu zrównoważeniu zbioru danych można połączyć metody oversampling oraz undersampling. Przykładem takich połączonych metod jest **SMOTE z ENN** (SMOTENN) oraz **SMOTE z Tomek links** (SMOTETOMEK) [1]. Tworzenie nowych próbek odbywa się tak jak metodzie SMOTE, a następnie usuwane są przykłady klasy większościowej według wybranego algorytmu (ENN lub Tomek links). Dodatkowo usuwane są sztuczne wygenerowane obserwacje, które za bardzo ingerują w przestrzeń klasy większościowej.

Rozdział 3

Przeprowadzone badania

3.1 Opis platformy i sposób realizacji badań

Wszystkie badania i testy zostały napisane z wykorzystaniem języka python oraz biblioteki scikit-learn, w której znajdują się implementacje użytych algorytmów klasyfikacji. Każdy test został przeprowadzony dla wszystkich zbiorów danych (rozdział 3.2), które zostały zaimportowane z plików. Po przetworzeniu danych, jeżeli wymagał tego algorytm, dane były poddawane procesowi standaryzacji. Następnie budowany był klasyfikator, który był oceniany z wykorzystaniem równomiernego sprawdzianu krzyżowego, dla $k = 10$. W kolejnym kroku oceniano klasyfikator z wykorzystaniem opisanych miar. Budowa klasyfikatora wraz z oceną były powtarzane 10 razy, a następnie wyniki zostały uśrednione. Końcowe wyniki były zapisywane do plików *.pdf* oraz *.tex*. Kod napisanych funkcji oraz testów został umieszczony na płycie CD. Do ich uruchomienia potrzebne są opisane w dalszej części pracy biblioteki. Nazwy wykorzystanych skryptów zostały wymienione dla każdego opisanego badania. Każdy test stanowi niezależną część i może być uruchamiany pojedynczo. Ponadto, na potrzeby przeprowadzonych testów, napisano:

- funkcje wczytujące dane z plików oraz wstępnie przetwarzające dane,
- sprawdzian krzyżowy,
- miary oceniające klasyfikator,
- funkcje prezentujące wyniki oraz zapisujące w postaci tabel do plików,
- klasyfikator ekspercki,
- meta-klasyfikator.

3.1.1 Język python

Język python to język programowania interpretowany, wysokiego poziomu, z dużą ilością dostępnych bibliotek. Python[22] posiada dynamiczne zarządzanie typami oraz automatyczne zarządzanie pamięcią. Wspiera kilka paradygmatów programowania, takich jak: obiektowy, imperatywny, funkcyjny i proceduralny. Został

zaprojektowany z troską o czytelność kodu oraz o składnię pozwalającą napisać program z mniejszą ilością kodu niż w językach C++ lub Java. Implementacja języka python dostępna jest na wiele systemów operacyjnych. Często wykorzystywany jest jako język skryptowy. Python jest projektem typu Open Source.

W pracy wykorzystano język python w wersji 2.7.11. Język ten wybrano ze względu na łatwość pisania w nim kodu, szybką możliwość nauki oraz szeroki wachlarz dostępnych bibliotek. Ważnym argumentem w wyborze były gotowe biblioteki z klasyfikatorami oraz do pracy z klasyfikacją danych. Dostępność bibliotek wizualizacyjnych dla tego języka, pozwoliła na przedstawienie wyników testów w formie graficznej. Napisane testy można w łatwy sposób rozbudować, zmodyfikować lub dodać nowe elementy.

3.1.2 Biblioteka scikit-learn

Scikit-learn[23] to proste i wydajne narzędzie do analizy i eksploracji danych. Jest to biblioteka uczenia maszynowego dla języka python. Rozpowszechnianie jej oparte jest na licencji BSD. W Scikt-learn zaimplementowane są (lub napisany jest kod obsługujący) różne algorytmy klasyfikacji, regresji, analizy skupień takie jak: maszyna wektorów nośnych, algorytmy najbliższego sąsiada, naiwny Bayes, drzewa decyzyjne, sieć neuronowa, zespoły klasyfikatorów. Z wykorzystaniem tej biblioteki można przygotować oraz przetworzyć odpowiednio dane. Możliwe jest także ocenianie oraz wizualizacja wyników.

W testach użyto biblioteki scikit-learn w wersji 0.18.1. Wykorzystano z niej algorytmy klasyfikacji oraz wstępnego przetwarzania danych.

3.1.3 Biblioteka imbalanced-learn

Biblioteka imbalanced-learn[24] zawiera zestaw narzędzi do wstępnego przetwarzania danych niezrównoważonych. Posiada ona zaimplementowane różne metody under- oraz over-sampling do równoważenia zbiorów danych. W pracy wykorzystano metody z biblioteki w wersji 0.2.1.

3.1.4 Pozostałe użyte biblioteki

Mlxtend

Mlxtend (machine learning extensions)[19] jest to biblioteka zawierająca różne narzędzia do pracy z danymi. W badaniach wykorzystano z niej algorytm Stacking.

Numpy

Numpy to pakiet umożliwiający obliczenia naukowe. Szczególnym elementem jest możliwość wykonywania obliczeń na tablicach N-wymiarowych.

Maptolib

Maptolib to biblioteka pythona, która tworzy różnego rodzaju wykresy 2D oraz interaktywne na różnych platformach.

Texttable

Texttable to prosty moduł napisany w języku python, służący do produkcji prostych tabel ASCII. Został wykorzystany do prezentacji wyników w konsoli.

Pylatex

Pylatex to biblioteka pythona służąca do tworzenia i kompilacji plików LaTeX. W pracy została wykorzystana do zapisu wyników badań w plikach .tex oraz .pdf.

3.2 Opis danych użytych w badaniach

Do przeprowadzenia badań użyto 26 różnych prawdziwych zbiorów danych (tabela 3.1) pochodzących z repozytorium serwisu „The UCI Machine Learning Repository" [21]. Dane wybrano ze względu na różnorodność typów danych, ilości rekordów, atrybutów oraz zróżnicowanie rozkładu klas. Większość z danych była używana w publikacjach podobnych tematycznie[10][15].

Nazwa danych	L. el.	Atrybuty	Rozkład klas	% kl. mn.	IR
abalone0_4	4177	8	4103/74	1.77	55.45
abalone041629	4177	8	3842/335	8.02	11.47
abalone16_29	4177	8	3916/261	6.25	15.0
balance_scale	625	4	576/49	7.84	11.76
breast_cancer	286	9	201/85	29.72	2.36
bupa	341	6	200/141	41.35	1.42
car	1728	6	1663/65	3.76	25.58
cmc	1473	9	1140/333	22.61	3.42
ecoli	336	7	301/35	10.42	8.6
german	1000	24	700/300	30.0	2.33
glass	214	9	197/17	7.94	11.59
haberman	306	3	225/81	26.47	2.78
heart_cleveland	303	13	268/35	11.55	7.66
hepatitis	155	19	123/32	20.65	3.84
horse_colic	368	22	232/136	36.96	1.71
ionosphere	351	34	225/126	35.9	1.79
new_thyroid	215	5	185/30	13.95	6.17
postoperative	90	8	66/24	26.67	2.75
seeds	210	7	140/70	33.33	2.0
solar_flare	1066	10	1023/43	4.03	23.79
transfusion	748	4	569/179	23.93	3.18
vehicle	846	18	647/199	23.52	3.25
vertebal	310	6	210/100	32.26	2.1
yeastME1	1484	8	1440/44	2.96	32.73
yeastME2	1484	8	1433/51	3.44	28.1
yeastME3	1484	8	1321/163	10.98	8.1

Tablica 3.1: Dane użyte w badaniach wraz z charakterystyką.

Wszystkie dane zostały zapisane w skrypcie, w folderze *praca/data/files*, a opis szczegółowy danych znajduje się w folderze *praca/data/files/data_ descrytpions*. Funkcje do importu danych znajdują się w pliku *praca/data/import_ data.py*. Do ogólnego importu danych z pliku, służy funkcja *importfile*, zaś wczytywanie danych użytych w projekcie odbywa się poprzez funkcje zaczynające się od *load_* . Import danych z pliku odbywa się z wykorzystaniem funkcji *genfromtext* oraz *load_ txt* z pakietu *numpy*. Atrybuty posiadające dane kategoryczne zapisane w postaci łańcuchów znaków, zostały zamienione na dane numeryczne. Cechy nominalne zostały zakodowane metodą *one hot encoding*. Dla danych zawierających więcej niż dwie klasy, klasa z najmniejszą liczebnością została wybrana jako klasa mniejszościowa, pozostałe klasy utworzyły klasę większościową. We wszystkich zbiorach danych, kategorie reprezentowane są w systemie binarnym. Pięć zbiorów danych posiadało brakujące wartości. Zostały one zastąpione wartościami środkowymi zbioru (medianą).

Analiza klas mniejszościowych

Dane użyte w badaniach poddano analizie sąsiedztwa, w celu określenia przynależności przykładów z klasy mniejszościowej do jednej z czterech grup. Analizę wykonano z wykorzystaniem algorytmu k najbliższych sąsiadów, dla $k = 5$. Do pomiaru odległości wykorzystano miarę Czebyszewa. Skrypt analizujący dane znajduje się w pliku *analyze_ db.py*. Przykłady klasyfikowano do grup w zależności od liczby sąsiadów[14]:

- safe - jeżeli w sąsiedztwie znajdowało się przynajmniej 4 przykłady z tej samej klasy,
- border - jeżeli liczba przykładów z obu klas była podobna, tj. dla 2-3 przykłady z tej samej klasy,
- rare - jeżeli w sąsiedztwie był tylko jeden przykład z tej klasy,
- outlier - jeżeli wszyscy sąsiedzi należeli do innej klasy.

Wyniki analizy przedstawiono w tabeli 3.2. Zbiory danych zostały posortowane od „najłatwiejszych" do „najtrudniejszych" w klasyfikacji.

	Safe [%]	Borderline [%]	Rare [%]	Outlier [%]
seeds	88.57	10.0	0.0	1.43
new_thyroid	73.33	10.0	6.67	10.0
vehicle	62.81	26.63	2.51	8.04
ionosphere	57.94	21.43	13.49	7.14
vertebal	56.0	33.0	2.0	9.0
yeastME3	50.92	32.52	10.43	6.13
yeastME1	36.36	52.27	0.0	11.36
ecoli	31.43	48.57	14.29	5.71
bupa	27.66	48.23	8.51	15.6
horse_colic	22.79	52.94	13.24	11.03
abalone0_4	37.84	32.43	18.92	10.81
german	9.0	51.33	14.67	25.0
breast_cancer	3.53	52.94	12.94	30.59
cmc	12.31	43.24	20.42	24.02
hepatitis	0.0	53.12	15.62	31.25
haberman	12.35	38.27	18.52	30.86
yeastME2	3.92	35.29	35.29	25.49
abalone041629	11.04	31.94	30.75	26.27
transfusion	13.41	36.31	21.79	28.49
car	1.54	33.85	35.38	29.23
glass	0.0	47.06	23.53	29.41
abalone16_29	3.45	31.8	34.1	30.65
solar_flare	4.65	16.28	46.51	32.56
heart_cleveland	0.0	5.71	48.57	45.71
balance_scale	0.0	22.45	30.61	46.94
postoperative	0.0	33.33	8.33	58.33

Tablica 3.2: Analiza przynależności przykładów z klasy mniejszościowej do grup. Dane posortowano w kolejności od najłatwiejszych w klasyfikacji do najtrudniejszych.

3.3 Ocena klasyfikatora w sprawdzianie krzyżowym k-krotnym.

W większości publikacji naukowych dotyczących klasyfikacji, ocena klasyfikatora mierzona jest z wykorzystaniem sprawdzianu krzyżowego (zwykle dla $k = 10$) oraz przedstawionych wcześniej miar. Jednakże, w tych publikacjach nie został opisany sposób obliczania współczynników dla sprawdzianu krzyżowego. Wykorzystanie różnych sposobów obliczania prowadzi do różnych wyników. Niektóre metody są mniej lub bardziej obciążone błędem. Różnice w wynikach, wynikające z przyjętej metody obliczeniowej, są szczególnie widoczne w sprawdzianie krzyżowym z losowym rozkładem danych oraz w klasyfikacji danych niezrównoważonych. Są dwie główne możliwości obliczania współczynników:

- obliczanie wartości współczynników dla każdej k-iteracji (klasyfikatora), a następnie obliczenie średniej z tych iteracji,
- stworzenie jednej wspólnej macierzy pomyłek dla każdej k-iteracji, a następnie obliczenie wskaźników.

W przypadku drugiego sposobu, poszczególne elementy macierzy pomyłek będą wynosić odpowiednio:

$$TP := \sum_{i=1}^{k} TP^{(i)}$$

$$FP := \sum_{i=1}^{k} FP^{(i)}$$

$$TN := \sum_{i=1}^{k} TN^{(i)}$$

$$FN := \sum_{i=1}^{k} FN^{(i)}$$

3.3.1 Test sposobów oceny klasyfikatora

W celu wyboru najlepszego sposobu oceny klasyfikatora, z najmniejszym błędem oraz wariancją, wykonano pięć porównujących testów dla różnych metod obliczania miar. Wszystkie testy miały takie same założenia oraz sposób wykonania. Testy wykonano na wygenerowanych losowo danych dla różnej liczby przykładów pozytywnych (od 1% do 10%). Jakość klasyfikacji była oceniana dla równomiernego sprawdzianu krzyżowego oraz dla losowego. Symulacje odbyły się w następujący sposób:

1. Wygeneruj zbiór 1500 losowych próbek z 2 atrybutami, 2 klasami o rozkładzie 4:1.
2. Wykonaj niezrównoważenie zbioru z ratio 0.1.
3. Wybierz m=[1..10]*10 przykładów klasy mniejszościowej oraz 1000-m przykładów klasy dominującej.
   (a) Wykonaj N iteracji:
      i. Wymieszaj dane.
      ii. Wykonaj sprawdzian krzyżowy k-krotny, k=10.
      iii. Oblicz współczynniki dla obu metod.
   (b) Oblicz odchylenie standardowe oraz średnie wartości współczynników.
4. Przedstaw wyniki odchylenia standardowego oraz średnie wartości współczynników dla różnego rozkładu klas.

Wykonanie testu N-krotnie (najkorzystniej n>100000) pozwala na obliczenie „prawdziwych" wartości miar oceny klasyfikacji. Powtórzenie sprawdzianu krzyżowego wielokrotnie pozwala na ocenę błędu oraz wariancji miar dla każdej metodyki. Przeprowadzenie testu dla danych zawierających tylko 1% obserwacji klasy mniejszościowej (przypadek ekstremalny, w zbiorze danych znajduje się wtedy tylko 10 takich przykładów) oznacza, że w niektórych iteracjach sprawdzianu krzyżowego nie będzie

przykładów poprawnie sklasyfikowanych z tej klasy. Brak dobrze sklasyfikowanych przykładów klasy mniejszościowej może mieć także miejsce w losowym sprawdzianie krzyżowym. Wynika to z braku równomiernego rozkładu obu klas.

Dokładność oraz błąd klasyfikatora

Dokładność klasyfikacji oraz błąd klasyfikatora, korzystając z metody pierwszej, będzie wynosić:

$$accuracy_{avg} := \frac{1}{k} \sum_{i=1}^{k} accuracy^{(i)}$$

a błąd klasyfikatora:

$$error\ rate_{avg} = 1 - accuracy_{avg}$$

Obliczając drugim sposobem, korzysta się z podstawowego wzoru z wykorzystaniem wspólnej macierzy pomyłek.

W przypadku dokładności oraz błędu klasyfikatora, niezależnie od przyjętej metodyki otrzymane wartości będą takie same, nieobciążone błędem.

Czułość, specyficzność, FPR oraz precyzja

Czułość, specyficzność, FPR oraz precyzja w metodzie pierwszej oblicza się według wzorów:

$$Sensitivity_{avg},\ Recall_{avg},\ TPR_{avg} := \sum_{i=1}^{k} TPR_{avg}^{(i)}$$

$$Specificity_{avg},\ TNR_{avg} := \sum_{i=1}^{k} TNR_{avg}^{(i)}$$

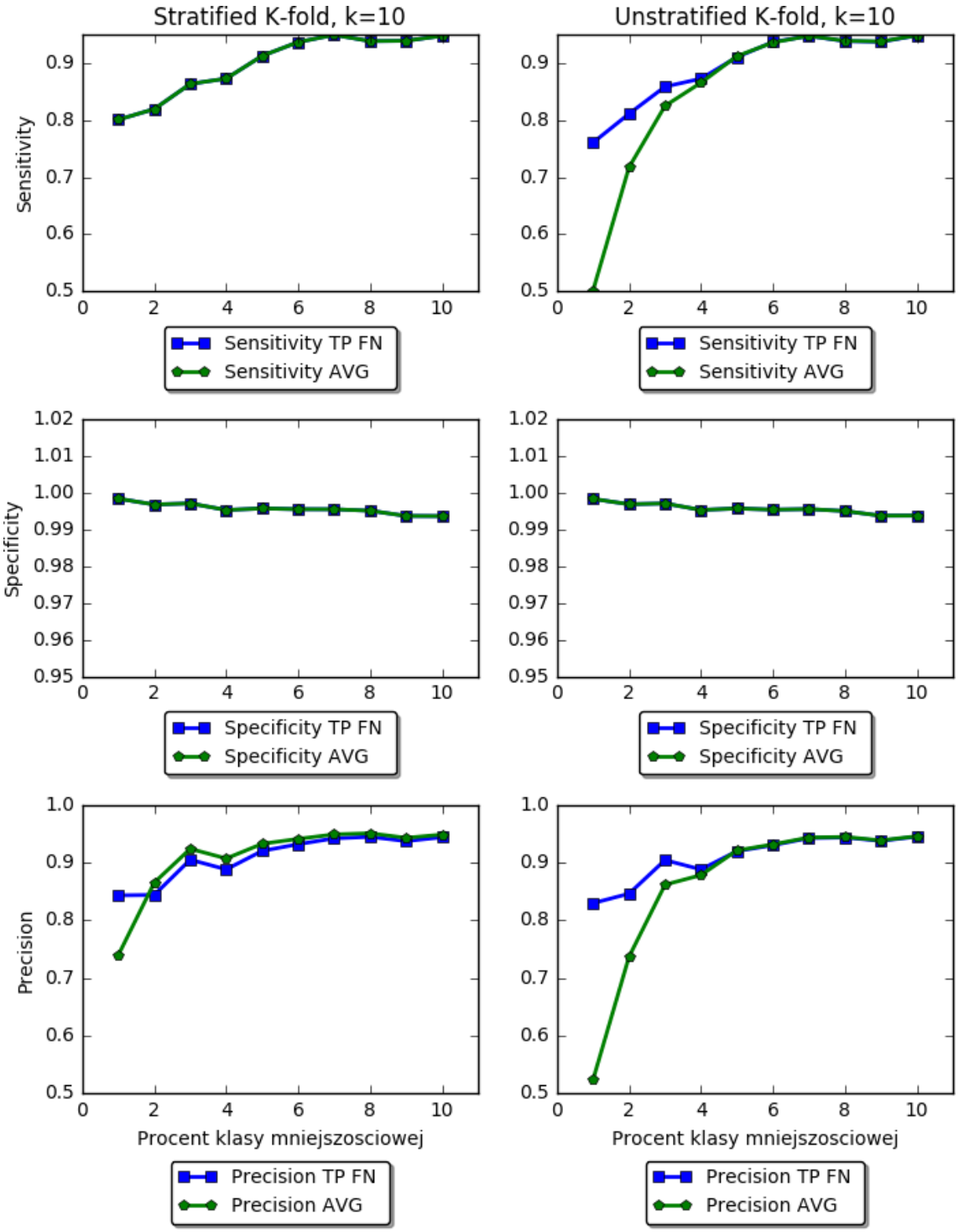
$$FPR_{avg} := \sum_{i=1}^{k} FPR_{avg}^{(i)}$$

$$Precision_{avg} := \sum_{i=1}^{k} Precision_{avg}^{(i)}$$

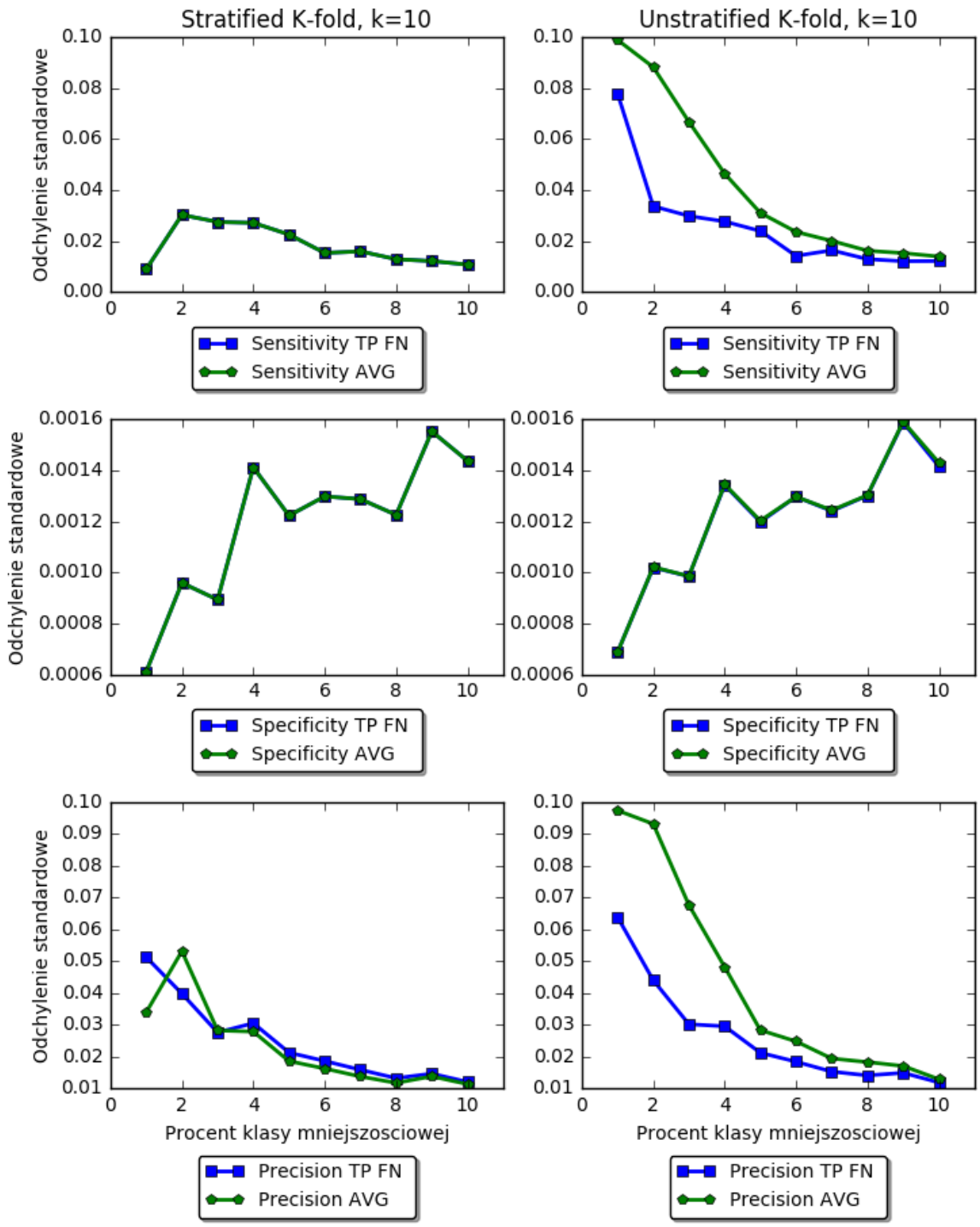
W drugim sposobie korzysta się z wspólnej macierzy pomyłek oraz z podstawowych wzorów.

Testy wyżej wymienionych współczynników zostały przeprowadzone z wykorzystaniem skryptu *test_ wsk.py*. Zauważono, że w przypadku równomiernego sprawdzianu krzyżowego, różnica w wynikach jest bardzo mała, poniżej 0.5%. Obie metody obarczone są małym błędem i wariancją. Natomiast w przypadku sprawdzianu krzyżowego z rozkładem losowym, metoda druga okazała się lepsza. Obliczona czułość oraz precyzja sposobem drugim, uzyskały wyniki z mniejszym błędem, bliższe wartości „prawdziwej". Natomiast specyficzność w obu metodach wyszła taka sama, ze względu na dużą liczbę przykładów z tej klasy. Sprawdzono, że w momencie odwrócenia liczebności klas, specyficzność posiada taką samą charakterystykę jak czułość. Różnice w wynikach obu sposobów zmniejszają się wraz ze wzrostem przykładów

klasy większościowej. Zazwyczaj przy 10% zawartości danych klasy zdominowanej w zbiorze, wyniki są takie same.



Rysunek 3.1: Wykresy czułości, specyficzności oraz precyzji w zależności od wielkości klasy mniejszościowej dla równomiernego oraz losowego sprawdzianu krzyżowego (k=10).



Rysunek 3.2: Odchylenie standardowe miar czułości, specyficzności oraz precyzji w zależności od wielkości klasy mniejszościowej dla równomiernego oraz losowego sprawdzianu krzyżowego (k=10).

Miara F_1

W niniejszej pracy oraz w wielu publikacjach, miara F-measure obliczana jest dla $\beta = 1$, dlatego testy przeprowadzono tylko dla tej wartości. Miarę F można obliczyć na trzy różne sposoby. Pierwszy polega na obliczeniu F dla każdego k klasyfikatora, a następnie uśrednienie wyników:

$$F_{avg} := \frac{1}{k} \sum_{i=1}^{k} F_1^{(i)}$$

Drugi sposób, to obliczenie średniej czułości i precyzji, a następnie miary F z podstawowego wzoru:

$$Pre_{avg} := \frac{1}{k} \sum_{i=1}^{k} Pre^{(i)}$$

$$Re_{avg} := \frac{1}{k} \sum_{i=1}^{k} Re^{(i)}$$

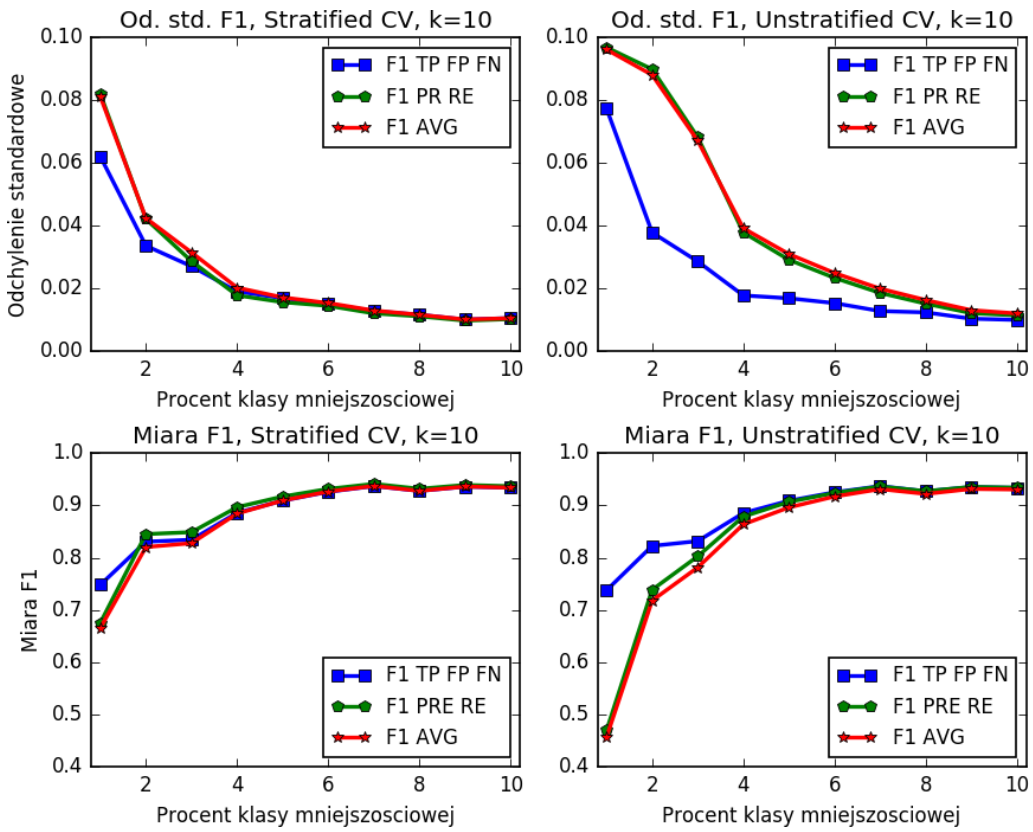
$$F_{pre,re} = 2 * \frac{Pre_{avg} * Re_{avg}}{Pre_{avg} + Re_{avg}}$$

Ostatni sposób, to obliczenie współczynnika F ze wspólnej, końcowej macierzy pomyłek:

$$F_{tp,fp,fn} = \frac{2 * TP}{2 * TP + FP + FN}$$

Do powyższych wzorów można dodać jeszcze sposób odrzucający oceny klasyfikatorów, dla których precyzja lub czułość są niezdefiniowane. Ta metoda została odrzucona ze względu na zawyżanie końcowej oceny.

Skrypt testujący powyższe trzy sposoby znajduje się w pliku *test_f1.py*. Analizując otrzymane wyniki (wykres 3.3), zauważono, że najbardziej powtarzalne wyniki otrzymano korzystając ze wzoru $F_{tp,fp,fn}$. W przypadku obu sprawdzianów krzyżowych, metoda ta generowała najmniejszy błąd. Niezależnie od ilości danych niezrównoważonych,wyniki otrzymane tą metodą różniły się nieznacznie, w przeciwieństwie do pozostałych sposobów. Podobnie jak w przypadku poprzednich miar, wraz ze wzrostem ilości danych niezrównoważonych, uzyskiwane wyniki były prawie takie same, niezależnie od sposobu obliczania.



Rysunek 3.3: Wykres odchylenia standardowego miar F1 oraz średniej miar F1, w zależności od wielkości klasy mniejszościowej dla równomiernego oraz losowego sprawdzianu krzyżowego (k=10).

Miara G-mean

Miara G-mean obliczona może zostać również na trzy różne sposoby. Pierwszy z nich to średnia z wszystkich klasyfikatorów:

$$G - mean_{avg} := \frac{1}{k} \sum_{i=1}^{k} G - mean^{(i)}$$

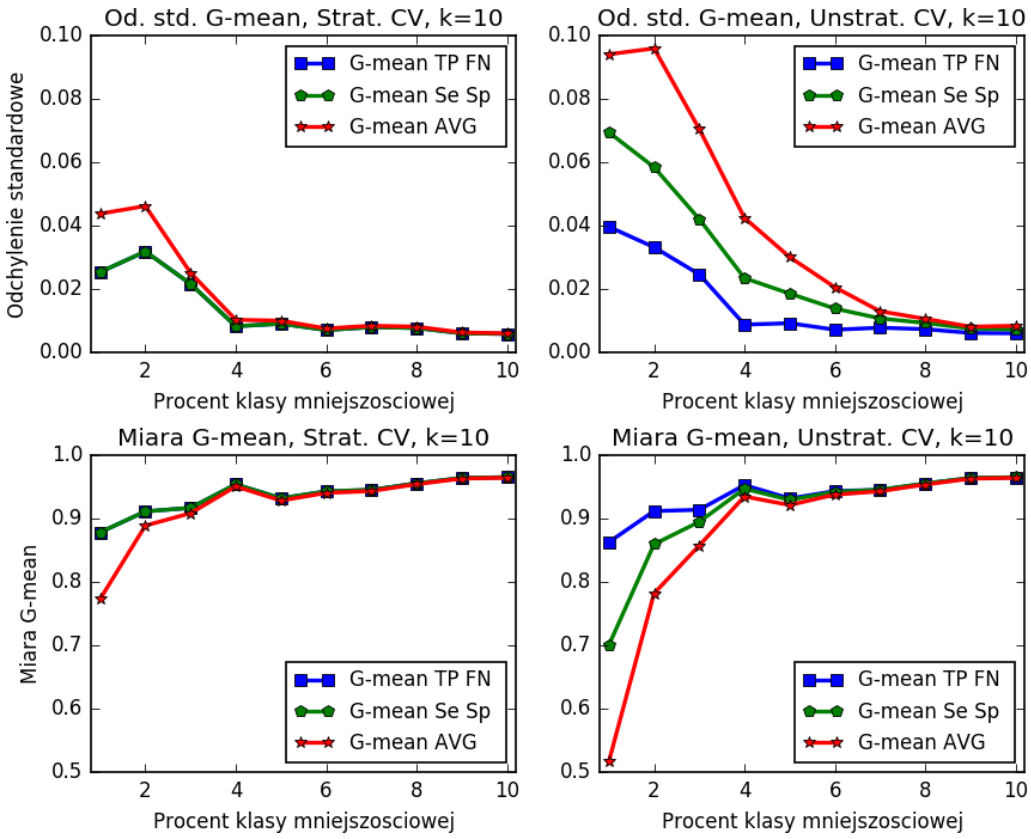
W drugim sposobie należy najpierw obliczyć średnią wartość czułości oraz specyficzności, a końcowy wynik G-mean oblicza się z głównego wzoru:

$$G - mean_{Se,Sp} = \sqrt{Sensitivity_{avg} * Specificity_{avg}}$$

W ostatniej metodzie obliczania G-mean, za czułość oraz specyficzność wstawia się właściwie wzory, a wartość oblicza się na podstawie zsumowanej macierzy pomyłek.

$$G - mean_{tp,fp,fn} = \sqrt{\frac{TP}{TP + FN} * \frac{TN}{TN + FP}}$$

Skrypt testujący powyższe wzory znajduje się w pliku *test_g_mean.py*. Analizując wyniki (wykres 3.4) równomiernego sprawdzianu krzyżowego, zaobserwowano, że wyniki $G - mean_{tp,fp,fn}$ oraz $G - mean_{Se,Sp}$ pokrywają się.



Rysunek 3.4: Wykres odchylenia standardowego miar G-mean oraz średniej miar G-mean w zależności, od wielkości klasy mniejszościowej, dla równomiernego oraz losowego sprawdzianu krzyżowego (k=10).

Natomiast w zwykłym sprawdzianie krzyżowym, pomiarem najmniej obarczonym błędem był $G - mean_{tp,fp,fn}$. Z wykorzystaniem tego wzoru, dla różnej zawartości klasy mniejszościowej w zbiorze danych otrzymano wyniki różniące się jedynie o kilka procent pomiędzy sobą, podczas gdy wyniki pozostałych metod różniły się aż o 15%-30%. Jednocześnie ta metoda dawała najmniejszy błąd odchylenia standardowego. Zauważono także, że w przypadku zawartości minimum 6% klasy mniejszościowej w danych, otrzymywane wyniki różnią się nieznacznie (poniżej 1%).

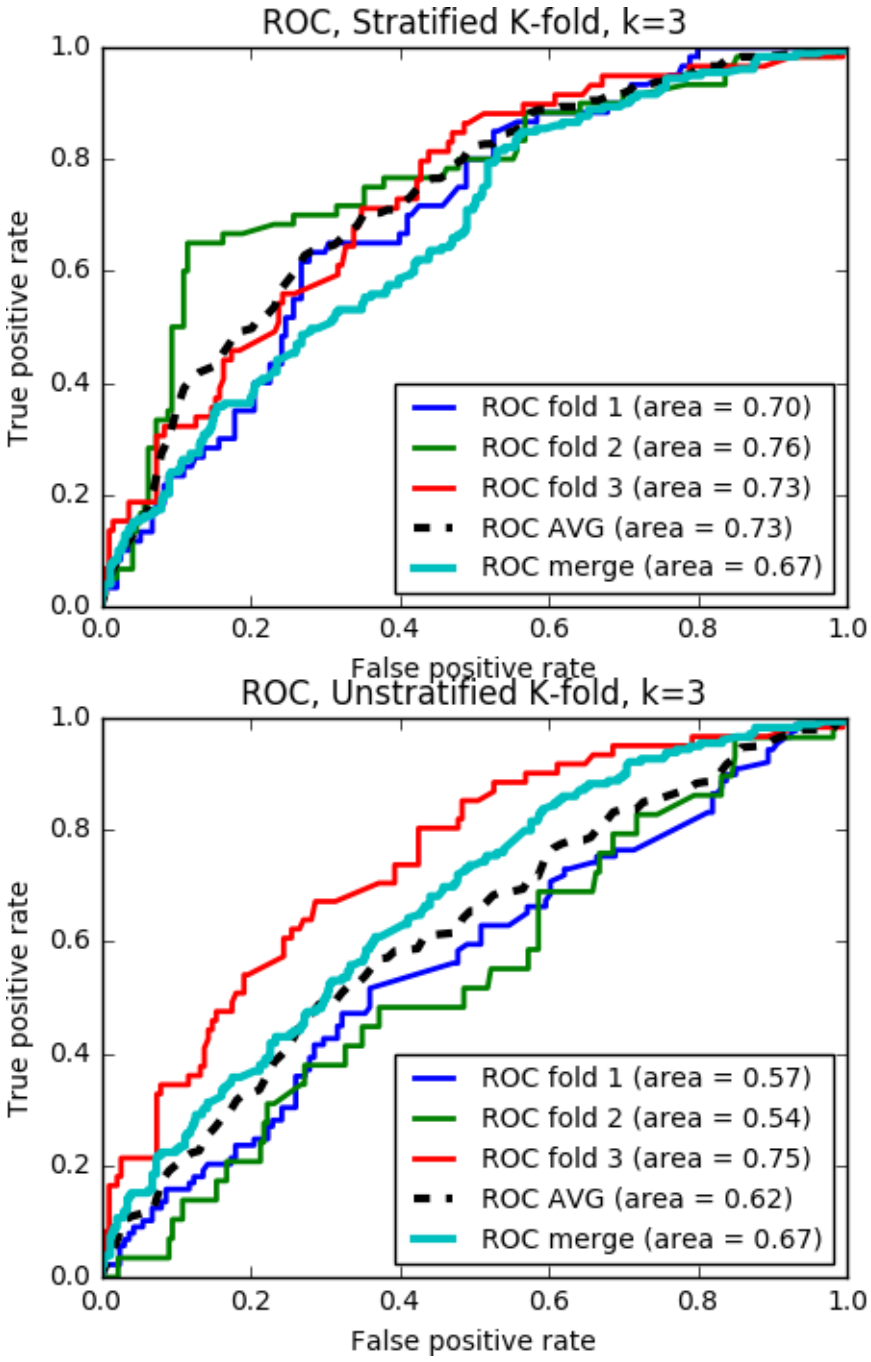
Krzywa ROC i miara AUC

Wartość AUC w sprawdzianie krzyżowym można obliczyć na dwa sposoby. W pierwszej metodzie konstruuje się krzywą ROC oraz oblicza się AUC dla każdego k klasyfikatora. Następnie oblicza się AUC_{AVG} poprzez obliczenie średniej:

$$AUC_{AVG} := \frac{1}{k} \sum_{i=1}^{k} AUC^{(i)}$$

W przypadku sprawdzianu krzyżowego z losowym rozkładem klas, może okazać się, że nie sklasyfikowano żadnego przykładu pozytywnego. Wtedy skonstruowanie krzywej ROC oraz obliczenie AUC będzie niemożliwe. W takich przypadkach podczas obliczania AUC_{AVG} można pominąć taki wynik.

Drugim sposobem jest połączenie prawdopodobieństwa przykładów testowych z każdej iteracji. Z połączonych obserwacji konstruuje się jedną krzywą ROC i oblicza AUC_{merge}. Korzystając z tego sposobu zakłada się, że klasyfikator ma dobrze skalibrowane określanie prawdopodobieństwa. Test obu metod zostały przeprowadzone z użyciem skryptu *test_roc.py* i danych *transfusion*. Wygenerowane krzywe (wykres 3.5) ROC k klasyfikatorów różnią się od siebie kształtem oraz powierzchnią AUC.



Rysunek 3.5: Wykresy krzywych ROC dla klasyfikatorów ze sprawdzianu krzyżowego równomiernego i normalnego wraz z obliczoną średnią ROC_{AVG} oraz ROC_{merge}.

W obu sprawdzianach krzyżowych obliczona średnia ROC_{AVG} oraz AUC_{AVG}

znajdują się pomiędzy wartościami otrzymanymi z klasyfikatorów cząstkowych (fold). Natomiast w sprawdzianie krzyżowym równomiernym wykres ROC_{merge} przez większość przebiegu znajduje się poniżej części składowych, a obliczona wartość AUC_{merge} jest niższa od AUC każdego klasyfikatora.

Podsumowanie

Analizując przeprowadzone testy, najlepsze wyniki osiągnęły miary obliczone metodami opartymi na zsumowanej macierzy pomyłek oraz na łączeniu wyników testów z każdej iteracji sprawdzianu krzyżowego. Obliczone w ten sposób współczynniki miały najbardziej stabilne wyniki, najmniejszy błąd oraz wariancję. Poniżej przedstawiono tabelę 3.3 z obliczonymi w różny sposób współczynnikami. Mimo równomiernego rozkładu klas, klasyfikator w części numer 2 nie rozpoznał ani jednego przykładu pozytywnego. W efekcie czułość oraz precyzja musiały zostać ustawione na 0, skutkiem czego miara G-mean oraz F1 dla tej części wynoszą 0. Pochodną tego zdarzenia jest zaniżenie wszystkich wartości średnich miar (oznaczonych w tabeli jako AVG). W dalszej części pracy, wszystkie miary w sprawdzianie krzyżowym będą obliczane na podstawie wspólnej macierzy pomyłek.

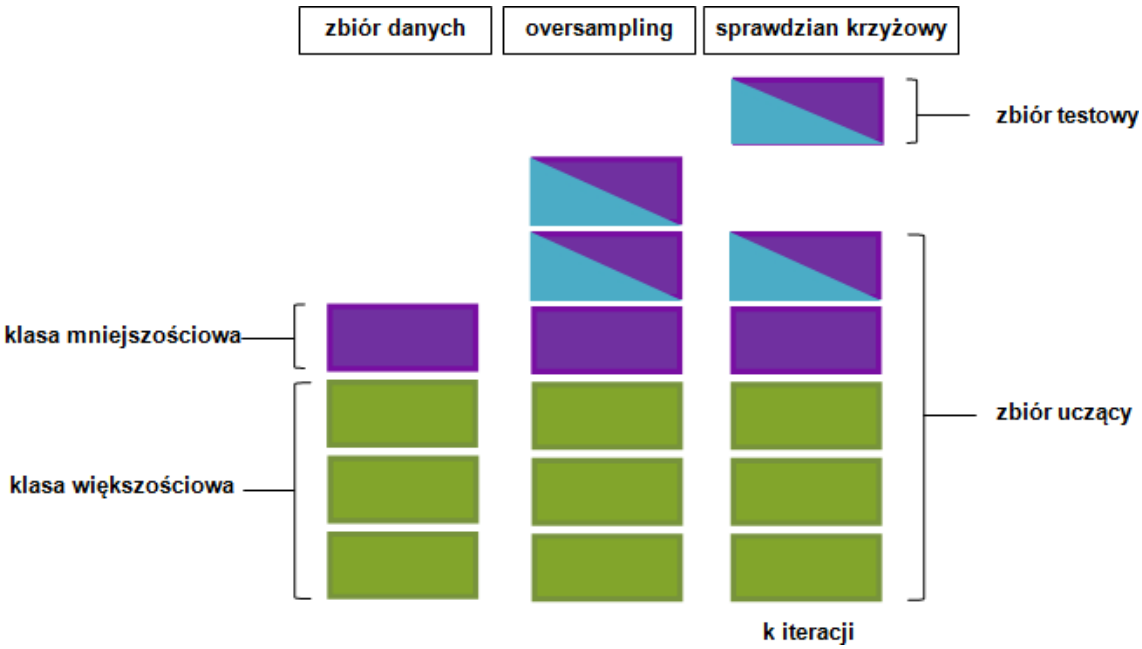
k-fold	Pos	Neg	TP	FP	FN	TN	Se	Sp	Pre	G-Mean	F_1
1	3	97	2	0	1	97	0,67	1,00	1,00	0,82	0,80
2	3	97	0	0	3	97	0,00	1,00	0,00	0,00	0,00
3	3	97	3	4	1	93	0,75	0,96	0,43	0,85	0,55
						AVG	**0,47**	**0,99**	**0,48**	**0,55**	**0,45**
						tp,fp,tn	**0,50**	**0,99**	**0,56**	**0,70**	**0,53**
										$\mathbf{G_{Se,Sp} = 0,68}$	$\mathbf{F_{Pre,Re} = 0,47}$

Tablica 3.3: Przykład obliczonych miar dla równomiernego sprawdzianu krzyżowego. Dla k=2, gdzie nie było pozytywnie sklasyfikowanych przykładów, wartości sensitivity, precision, F_1 zostały ustawione na 0, aby uniknąć dzielenia przez zero. W wierszu oznaczonym jako „tp,fp,tn", wskaźniki zostały obliczone na podstawie wspólnej macierzy pomyłek.

3.4 Równoważenie liczebności klas w danych w klasyfikacji ze sprawdzianem krzyżowym

Równoważenie liczebności klas w zbiorze danych wykonuje się w celu poprawy klasyfikacji klasy mniejszościowej. Istnieją różne techniki balansowania zbiorów (opisanych w rozdziale 2.2). Są to między innymi usuwanie przykładów z klasy większościowej oraz dodawanie nowych obserwacji z klasy mniejszościowej. Oceniając klasyfikator z wykorzystaniem sprawdzianu krzyżowego, bilansowanie zbiorów można wykonać przed sprawdzianem krzyżowym oraz w trakcie trwania sprawdzianu, każdorazowo po stworzeniu zbioru uczącego. W przypadku generowania nowych obserwacji (oversampling), metoda pierwsza może prowadzić do nadmiernego dopasowania. Sprawdzian krzyżowy k-krotny, dzieli główny zbiór danych na k części i buduje klasyfikator w oparciu $k-1$ części. Następnie testuje go w oparciu o część numer

k. Może wystąpić przypadek, w którym ocena klasyfikatora będzie odbywała się w oparciu o sztuczne przykłady, które były wygenerowane na podstawie danych tworzących zbiór uczący. Takie zdarzenie wypacza cel wykonywania sprawdzianu krzyżowego, który polega na wykonywaniu uczenia i testowania na różnych danych. Wykonanie bilansowania przed sprawdzianem krzyżowym może prowadzić do nadmiernego dopasowania klasyfikatora oraz do zawyżenia oceny.



Rysunek 3.6: Przykład sprawdzianu krzyżowego z wykonanym oversamplingiem przed sprawdzianem. Sztucznie wygenerowane dane (niebiesko-fioletowy prostokąt) zostały użyte jako zbiór testowy.

Celem testu było sprawdzenie, którą metodą można otrzymać bardziej wiarygodne wyniki. Aby sprawdzić obie metody, każdy początkowy zbiór danych został podzielony na zbiór testowy (zawierający 80% wszystkich przykładów) oraz na zbiór walidacyjny (20%). Wykonano zbilansowany podział danych, tj. każdy zbiór zawierał proporcjonalnie taką samą liczbę obserwacji obu klas. Zbiór walidacyjny został wydzielony w celu poddania dodatkowej ocenie każdego k klasyfikatora. Mając dodatkowy zbiór walidacyjny, który jest nieznany zbudowanemu modelowi, można sprawdzić, w jakim stopniu poprawnie generalizuje on dane uczące oraz porównać otrzymane wyniki z wynikami sprawdzianu krzyżowego. W kolejnym etapie, dla pierwszego sposobu wygenerowano nowe sztuczne próbki klasy mniejszościowej algorytmem oversampling SMOTE dla całego zbioru danych przed podziałem, a następnie poddano ocenie klasyfikator z wykorzystaniem sprawdzianu krzyżowego oraz dodatkowego zbioru walidacyjnego w każdej iteracji. Natomiast w drugim sposobie, dodatkowe sztuczne próbki generowane były w trakcie sprawdzianu krzyżowego tylko dla k zbioru uczącego dopiero po utworzeniu k zbioru treningowego i k zbioru testowego. W efekcie żaden zbiór testowy nie zawierał sztucznych przykładów. Dodatkowa walidacja odbywała się tak samo jak w pierwszym sposobie. Badanie wykonano

na prawdziwych danych (opisanych w tej pracy), z wykorzystaniem klasyfikatorów: drzewa decyzyjnego, kNN, naiwnego klasyfikatora Bayesa oraz SVM. Przedstawiony poniżej test znajduje się w pliku *cross_ val_ oversampling.py*. Ze względu na dużą objętość wyników, w pracy przedstawiono tylko dane dla drzewa decyzyjnego.

Analizując otrzymane wyniki (tabela 3.4 i 3.5) zauważono bardzo dobre wyniki klasyfikatora dla danych poddanych oversamplingowi przed sprawdzianem krzyżowym. Niestety, badając klasyfikator nieznanym dla niego zbiorem walidacyjnym,

	Decision Tree				Decision Tree TEST				
	Sp	F1	G	AUC	Sp	F1	G_t	AUC	$G - G_t$
abalone0_4	0.99	0.99	0.99	0.99	0.79	0.58	0.88	0.88	0.11
abalone041629	0.92	0.91	0.9	0.9	0.5	0.34	0.66	0.69	0.24
abalone16_29	0.93	0.92	0.92	0.92	0.51	0.34	0.68	0.71	0.24
balance_scale	0.92	0.9	0.9	0.9	0.04	0.04	0.19	0.47	0.71
breast_cancer	0.74	0.74	0.74	0.75	0.47	0.45	0.59	0.61	0.15
bupa	0.69	0.66	0.65	0.65	0.57	0.55	0.6	0.6	0.05
car	1.0	1.0	1.0	1.0	0.9	0.95	0.95	0.95	0.05
cmc	0.8	0.81	0.81	0.81	0.39	0.37	0.55	0.6	0.26
ecoli	0.93	0.93	0.92	0.93	0.81	0.62	0.86	0.86	0.06
german	0.74	0.75	0.75	0.75	0.44	0.44	0.58	0.59	0.17
glass	0.96	0.93	0.93	0.93	0.56	0.5	0.73	0.75	0.2
haberman	0.72	0.73	0.74	0.74	0.35	0.35	0.52	0.56	0.22
heart_cleveland	0.88	0.87	0.86	0.86	0.24	0.29	0.47	0.59	0.39
hepatitis	0.92	0.86	0.84	0.85	0.39	0.33	0.54	0.57	0.3
horse_colic	0.84	0.82	0.82	0.82	0.65	0.66	0.73	0.73	0.09
ionosphere	0.88	0.88	0.88	0.88	0.87	0.81	0.86	0.86	0.02
new_thyroid	0.96	0.96	0.96	0.96	1.0	0.95	0.99	0.99	-0.03
postoperative	0.68	0.7	0.71	0.71	0.33	0.34	0.51	0.56	0.2
seeds	0.97	0.96	0.96	0.96	0.62	0.7	0.76	0.77	0.2
solar_flare	0.96	0.96	0.96	0.97	0.13	0.19	0.36	0.63	0.6
transfusion	0.76	0.75	0.75	0.77	0.4	0.35	0.54	0.58	0.21
vehicle	0.89	0.92	0.92	0.92	0.72	0.76	0.83	0.83	0.09
vertebal	0.89	0.86	0.85	0.85	0.73	0.66	0.75	0.75	0.1
yeastME1	0.99	0.99	0.99	0.99	0.72	0.68	0.84	0.85	0.15
yeastME2	0.98	0.96	0.96	0.96	0.69	0.42	0.81	0.82	0.15
yeastME3	0.96	0.96	0.96	0.96	0.66	0.65	0.79	0.81	0.17

Tablica 3.4: Wyniki sprawdzianu krzyżowego (metoda pierwsza) drzewa decyzyjnego z oversampling SMOTE. Dane w kolumnach „Decision Tree TEST" to wyniki otrzymane na podstawie zbioru walidacyjnego. Ostatnia kolumna zawiera różnicę miar G ze sprawdzianu krzyżowego i zbioru walidacyjnego $G - G_T$ (im bliżej zera tym lepiej).

otrzymane rezultaty były zdecydowanie gorsze. Porównując otrzymane wyniki ze sprawdzianu krzyżowego oraz z nieznanego klasyfikatorowi zbioru walidacyjnyjego, spostrzeżono znaczne zawyżanie wyników (dla miary G-mean wyniki jest średnio wyższy o 0.20, tak samo w stosunku do miary G-mean z metody drugiej). Świadczy to o nadmiernym dopasowaniu klasyfikatora do danych i o zawyżaniu wyników jakości klasyfikacji przez tę metodę. W drugiej metodzie, wyniki pomiarów pomiędzy sprawdzianem krzyżowym oraz zbiorem walidacyjnym nie różnią się tak bardzo. Dla miary G-mean, w większości baz, otrzymany wyniki z sprawdzianu krzyżowego jest nieznacznie niższy od wyniku ze zbioru walidacyjnego.

Wnioskiem wynikającym z powyższego badania jest konieczność stosowania over-

samplingu danych w trakcie sprawdzianu krzyżowego, tak aby zbiór testowy nie zawierał sztucznie wygenerowanych próbek. Ocena klasyfikacji tą metodą, daje najbardziej wiarygodne wyniki.

	Decision Tree				Decision Tree TEST				
	Sp	F1	G	AUC	Sp	F1	G_t	AUC	$G - G_t$
abalone0_4	0.68	0.54	0.82	0.83	0.77	0.58	0.87	0.88	-0.05
abalone041629	0.42	0.31	0.61	0.65	0.45	0.32	0.63	0.67	-0.02
abalone16_29	0.39	0.27	0.59	0.64	0.52	0.34	0.68	0.71	-0.09
balance_scale	0.03	0.02	0.15	0.47	0.04	0.03	0.19	0.46	-0.04
breast_cancer	0.41	0.4	0.55	0.57	0.41	0.4	0.55	0.57	0.0
bupa	0.54	0.54	0.6	0.6	0.55	0.56	0.62	0.62	-0.02
car	0.98	0.98	0.99	0.99	0.91	0.95	0.95	0.95	0.04
cmc	0.38	0.36	0.55	0.59	0.39	0.37	0.56	0.61	-0.01
ecoli	0.54	0.53	0.71	0.74	0.62	0.58	0.76	0.78	-0.05
german	0.55	0.54	0.66	0.67	0.5	0.49	0.62	0.63	0.04
glass	0.14	0.16	0.37	0.54	0.56	0.56	0.73	0.76	-0.36
haberman	0.49	0.42	0.58	0.59	0.4	0.36	0.53	0.55	0.05
heart_cleveland	0.43	0.35	0.61	0.65	0.24	0.29	0.48	0.59	0.13
hepatitis	0.5	0.52	0.67	0.69	0.72	0.58	0.77	0.77	-0.1
horse_colic	0.78	0.77	0.82	0.82	0.65	0.66	0.73	0.73	0.09
ionosphere	0.82	0.81	0.85	0.85	0.85	0.81	0.85	0.85	0.0
new_thyroid	0.88	0.86	0.92	0.92	0.94	0.87	0.95	0.95	-0.03
postoperative	0.32	0.27	0.44	0.48	0.33	0.3	0.47	0.48	-0.03
seeds	0.95	0.91	0.94	0.94	0.62	0.68	0.75	0.76	0.19
solar_flare	0.21	0.2	0.45	0.62	0.13	0.18	0.36	0.63	0.09
transfusion	0.36	0.34	0.52	0.58	0.39	0.35	0.54	0.57	-0.02
vehicle	0.82	0.82	0.88	0.88	0.76	0.79	0.85	0.85	0.03
vertebal	0.66	0.68	0.75	0.76	0.63	0.63	0.72	0.73	0.03
yeastME1	0.63	0.59	0.79	0.81	0.72	0.69	0.84	0.86	-0.05
yeastME2	0.29	0.2	0.53	0.62	0.41	0.3	0.63	0.68	-0.1
yeastME3	0.81	0.77	0.88	0.89	0.68	0.68	0.81	0.82	0.07

Tablica 3.5: Wyniki sprawdzianu krzyżowego (metoda druga) drzewa decyzyjnego z oversampling SMOTE. Dane w kolumnach „Decision Tree TEST" to wyniki otrzymane na podstawie zbioru walidacyjnego. Ostatnia kolumna zawiera różnicę miar G ze sprawdzianu krzyżowego i zbioru walidacyjnego $G - G_T$ (im bliżej zera tym lepiej).

Rozdział 4

Meta-metody

4.1 Bagging

Klasyfikator bagging to meta-klasyfikator, który trenuje n klasyfikatorów bazowych na losowych podzbiorach oryginalnego zbioru danych, a następnie poprzez głosowanie lub uśrednianie indywidualnych prognoz nadaje ostateczną klasę. W procesie tworzenia klasyfikatora bagging, należy wybrać klasyfikator bazowy oraz określić liczbę n tworzonych instancji tego klasyfikatora. Można także zdefiniować, czy losowe podzbiory mają być tworzone próbą boostrap, maksymalną liczebność podzbiorów oraz liczebność atrybutów. Zmieniając liczbę estymatorów, liczebność podzbiorów oraz atrybutów wpływa się na jakość klasyfikacji. Badania przeprowadzono z wykorzystaniem trzech różnych klasyfikatorów bazowych tj. z drzewem decyzyjnym, naiwnym klasyfikatorem bayesowskim, klasyfikatorem k najbliższych sąsiadów oraz dla różnych wartości przykładów oraz atrybutów. Każdy test został przeprowadzony z wykorzystaniem 5, 10, 15, 30, 50, 100 i 200 estymatorów. Wszystkie podzbiory zostały utworzone z wykorzystaniem próby boostrap, zatem podzbiory z taką samą liczebnością jak zbiór główny, będą różniły się od siebie próbkami.

4.1.1 Bagging z naiwnym klasyfikatorem bayesowskim

Zbudowano klasyfikator bagging z naiwnym klasyfikatorem Bayesa (NKB). Test ten znajduje się w pliku *bagging_NB.py*. W pierwszym etapie badań wybrano standardowe ustawienia, czyli liczebność podzbiorów (*max_samples*) oraz atrybutów (*max_features*) była taka sama jak w zbiorze oryginalnym. W tabelach poniżej przedstawiono dokładność klasyfikacji (tabela 4.1) oraz specyficzność klasy mniejszościowej (tabela 4.2). W kolumnie drugiej znajdują się wyniki dla samego naiwnego klasyfikatora Bayesa, natomiast w kolejnych kolumnach znajdują się wyniki meta-klasyfikatora bagging z różną liczbą modeli bazowych (dla 5, 10, 15, 30, 50, 100, 200 klasyfikatorów). Analizując otrzymane wyniki, zauważono tylko minimalny wzrost (około 1%) poprawy klasyfikacji obu klas dla połowy zbiorów danych. Dla prawie wszystkich zbiorów danych wartość miary G-mean zmieniła się minimalnie,

poniżej błędu. Test został wykonany wielokrotnie, a otrzymywane wyniki różniły się w bardzo niewielkim stopniu.

Zbiór danych	NKB	5	10	15	30	50	100	200
seeds	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**
new_thyroid	0.96	0.96	**0.97**	**0.97**	**0.97**	0.96	0.96	0.96
vehicle	0.66	**0.67**	**0.67**	**0.67**	**0.67**	0.66	0.66	0.66
ionosphere	**0.87**	0.85	0.85	0.85	0.86	0.86	**0.87**	**0.87**
vertebal	**0.78**	0.77	0.77	0.77	0.77	0.77	**0.78**	0.77
yeastME3	**0.27**	0.17	0.21	0.23	0.25	0.25	0.25	0.25
ecoli	0.78	0.77	0.79	**0.8**	0.79	0.79	0.78	0.79
bupa	0.54	0.53	**0.55**	0.54	**0.55**	**0.55**	0.54	0.54
horse_colic	**0.78**	0.76	0.77	0.77	0.77	0.77	**0.78**	0.77
german	**0.73**	**0.73**	**0.73**	**0.73**	0.72	0.72	0.72	0.72
breast_cancer	**0.72**	**0.72**	0.71	0.71	**0.72**	**0.72**	**0.72**	**0.72**
cmc	**0.68**	**0.68**	**0.68**	**0.68**	**0.68**	**0.68**	**0.68**	**0.68**
hepatitis	0.66	0.65	0.65	0.66	**0.68**	**0.68**	**0.68**	**0.68**
haberman	0.73	**0.74**	**0.74**	**0.74**	**0.74**	**0.74**	**0.74**	**0.74**
transfusion	**0.74**	**0.74**	**0.74**	**0.74**	**0.74**	**0.74**	**0.74**	**0.74**
car	0.89	**0.91**	0.9	0.9	0.9	0.9	0.9	0.9
glass	0.48	0.48	0.49	**0.53**	0.48	0.49	0.49	0.5
abalone16_29	0.68	0.68	0.68	**0.69**	0.68	0.68	0.68	0.68
solar_flare	**0.65**	0.62	0.61	0.63	0.62	0.62	0.62	0.63
heart_cleveland	**0.81**	0.8	**0.81**	0.8	**0.81**	**0.81**	0.8	0.8
balance_scale	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**
postoperative	**0.67**	0.62	0.64	0.62	0.6	0.61	0.62	0.63

Tablica 4.1: Dokładność klasyfikatora bagging NKB, dla *max_features = 1.0* oraz *max_samples = 1.0.*

Zbiór danych	NKB	5	10	15	30	50	100	200
seeds	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**
new_thyroid	**0.87**	**0.87**	**0.87**	**0.87**	**0.87**	**0.87**	**0.87**	**0.87**
vehicle	0.84	0.83	0.84	0.84	**0.85**	0.84	0.84	0.84
ionosphere	0.76	0.76	**0.77**	**0.77**	**0.77**	0.76	0.76	0.76
vertebal	**0.87**	0.86	0.86	0.86	0.86	0.86	**0.87**	0.86
yeastME3	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
ecoli	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**
bupa	0.74	0.75	**0.77**	**0.77**	0.74	0.75	0.75	0.74
horse_colic	**0.75**	**0.75**	0.74	**0.75**	**0.75**	0.74	0.74	0.73
german	0.62	0.6	0.6	0.6	0.63	0.63	**0.65**	**0.65**
breast_cancer	0.44	0.45	0.44	0.44	**0.46**	**0.46**	**0.46**	**0.46**
cmc	0.61	0.61	0.61	0.61	**0.62**	**0.62**	**0.62**	0.61
hepatitis	**0.78**	0.72	0.72	0.75	0.75	0.75	0.75	0.75
haberman	0.17	**0.2**	0.19	0.19	**0.2**	**0.2**	**0.2**	**0.2**
transfusion	0.2	**0.21**	**0.21**	**0.21**	**0.21**	**0.21**	**0.21**	**0.21**
car	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
glass	0.82	0.82	0.82	0.82	**0.88**	**0.88**	**0.88**	0.82
abalone16_29	0.58	**0.59**	0.58	0.58	0.58	0.58	0.58	0.58
solar_flare	**0.93**	**0.93**	**0.93**	**0.93**	**0.93**	**0.93**	**0.93**	**0.93**
heart_cleveland	**0.63**	0.57	0.6	0.57	0.6	0.57	0.57	0.57
balance_scale	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
postoperative	0.17	**0.21**	**0.21**	**0.21**	0.12	0.12	0.12	0.12

Tablica 4.2: Specyficzność klasy mniejszościowej, dla klasyfikatora bagging NKB i parametrów: *max_features = 1.0* oraz *max_samples = 1.0.*

W kolejnym teście znajdującym się w pliku *gridsearch/bagging_NB.py*, wykonano zachłanne obliczenia polegające na wyłonieniu najlepszych ustawień

klasyfikatora maksymalizującego miarę F_1 klasy mniejszościowej. W tym celu wykonano wyszukiwanie zachłanne najlepszych parametrów (liczby atrybutów oraz liczebności zbiorów) spośród *max_features=[0.4, 0.6, 0.7, 0.8, 0.9, 1.0]* oraz *max_samples=[0.4, 0.6, 0.7, 0.8, 0.9, 1.0]*. Wyszukanie wykonano dla n-klasyfikatorów ze zbioru *[5, 10, 15, 30, 50, 100, 200]* oraz dla każdego zbioru danych. Obliczona średnia wartość parametru *max_features* wyniosła 0.72, a *max_samples* 0.68, zaś mediana dla obu wartości to 0.7.

Zbudowany meta-klasyfikator bagging z parametrami *max_features=0.72* i *max_samples=0.68* poradził sobie lepiej z klasyfikacją niż klasyfikator bagging z domyślnymi parametrami oraz zwykły klasyfikator. W przypadku 17 zbiorów danych osiągnął lepszą dokładność (tabela 4.3). Dokładność klasyfikacji wzrosła średnio o 2-3%, a w 3 zbiorach wzrost był zdecydowanie większy. Natomiast w 5 zbiorach danych, klasyfikator bagging uzyskał taką samą lub minimalne gorszą dokładność. W zespołach liczących powyżej 10 klasyfikatorów wystąpił minimalny przyrost dokładności poniżej 1%. Prawie dla wszystkich danych, zwiększyła się czułość klasy większościowej, kosztem specyficzności (tabela 4.4) klasy mniejszościowej. Podobnie jak dla poprzedniego klasyfikatora, miara G-mean minimalnie spadła lub wzrosła w stosunku do NKB.

Zbiór danych	NKB	5	10	15	30	50	100	200
seeds	0.9	0.89	0.9	**0.91**	**0.91**	**0.91**	0.9	0.9
new_thyroid	0.96	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**
vehicle	0.66	0.68	0.68	0.67	**0.69**	0.67	0.67	0.68
ionosphere	**0.87**	0.83	0.86	0.83	**0.87**	**0.87**	0.86	0.86
vertebal	**0.78**	0.75	0.76	0.77	0.77	0.77	0.77	**0.78**
yeastME3	0.27	**0.34**	0.25	0.17	0.21	0.23	0.23	0.22
ecoli	0.78	0.83	0.8	0.8	0.81	0.83	**0.84**	0.83
bupa	0.54	0.57	0.57	**0.6**	**0.6**	0.59	0.59	**0.6**
horse_colic	0.78	0.73	0.76	0.73	0.76	0.77	0.78	**0.79**
german	**0.73**	0.66	0.66	0.68	0.7	0.72	0.72	**0.73**
breast_cancer	0.72	**0.73**	**0.73**	**0.73**	**0.73**	**0.73**	**0.73**	**0.73**
cmc	0.68	**0.72**	0.71	0.71	0.71	0.71	**0.72**	**0.72**
hepatitis	0.66	0.63	0.67	**0.68**	0.66	0.66	**0.68**	0.67
haberman	0.73	0.74	**0.75**	**0.75**	0.74	0.74	0.74	0.74
transfusion	0.74	0.76	0.75	0.75	0.75	**0.77**	**0.77**	**0.77**
car	0.89	**0.92**	0.9	0.9	0.9	0.9	0.9	0.9
glass	0.48	**0.76**	0.59	0.61	0.59	0.6	0.6	0.59
abalone16_29	0.68	0.8	**0.81**	0.8	0.79	0.79	0.79	0.79
solar_flare	**0.65**	0.6	0.53	0.52	0.64	0.58	0.53	0.58
heart_cleveland	0.81	0.78	0.8	0.81	0.82	0.82	**0.83**	0.82
balance_scale	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**
postoperative	0.67	**0.69**	0.63	0.62	0.64	0.62	0.63	0.64

Tablica 4.3: Dokładność klasyfikatora bagging NKB, dla *max_features = 0.72* oraz *max_samples = 0.68*.

Zbiór danych	NKB	5	10	15	30	50	100	200
seeds	**0.91**	0.89	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**
new_thyroid	**0.87**	0.8	**0.87**	**0.87**	**0.87**	**0.87**	**0.87**	**0.87**
vehicle	0.84	**0.85**	0.82	0.83	0.82	0.82	0.82	0.83
ionosphere	0.76	**0.83**	0.75	0.79	0.75	0.73	0.72	0.74
vertebal	**0.87**	0.81	0.83	0.86	0.86	0.86	0.84	0.86
yeastME3	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
ecoli	**0.94**	0.83	0.86	0.8	0.91	0.91	0.91	0.91
bupa	**0.74**	0.5	0.63	0.62	0.64	0.7	0.67	0.67
horse_colic	0.75	0.72	0.74	0.75	0.75	0.75	**0.76**	**0.76**
german	0.62	**0.73**	0.71	0.69	0.69	0.64	0.63	0.64
breast_cancer	**0.44**	0.39	**0.44**	**0.44**	**0.44**	**0.44**	0.42	0.42
cmc	**0.61**	0.51	0.53	0.5	0.52	0.5	0.48	0.5
hepatitis	0.78	**0.84**	0.75	0.75	0.72	0.72	0.75	0.75
haberman	0.17	0.1	**0.19**	0.16	0.17	0.16	0.14	0.14
transfusion	**0.2**	0.15	0.16	0.17	0.15	0.16	0.16	0.16
car	**1.0**	0.45	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
glass	**0.82**	0.59	0.76	0.71	0.71	0.71	0.76	0.76
abalone16_29	**0.58**	0.28	0.4	0.42	0.43	0.43	0.44	0.43
solar_flare	**0.93**	0.81	**0.93**	**0.93**	**0.93**	**0.93**	**0.93**	**0.93**
heart_cleveland	**0.63**	0.57	0.54	0.49	0.46	0.46	0.51	0.46
balance_scale	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
postoperative	**0.17**	**0.17**	0.12	0.12	**0.17**	0.12	0.08	0.12

Tablica 4.4: Specyficzność klasy mniejszościowej, dla klasyfikatora bagging NKB i parametrów: *max_features = 0.72* oraz *max_samples = 0.68*.

4.1.2 Bagging drzewa decyzyjne

Do kolejnych testów z meta-klasyfikatorem bagging wybrano drzewo decyzyjne. Budując klasyfikator drzewa decyzyjnego można zdefiniować maksymalną głębokość drzewa. Również meta-klasyfikator bagging można zbudować z drzew o różnej maksymalnej głębokości. Do testów wybrano drzewo bez ograniczenia głębokości oraz drzewa z ograniczeniami do maksymalnie 3, 5, 7, 10, 15 i 20 poziomu. Meta-klasyfikator był budowany z 5, 10, 20 lub 50 klasyfikatorów. Dla porównania, w wynikach zamieszczono pojedyncze drzewo decyzyjne o różnej głębokości. Przeprowadzony test znajduje się w pliku *bagging_tree.py*. Ze względu na dużą objętość tabel pominięto wyniki niektórych zbiorów danych. Dla bazy *seeds* i *new_thyroid*, dokładność klasyfikacji i wykrywalność klas pozostała na takim samym poziomie niezależnie od głębokości drzewa i liczby klasyfikatorów. Natomiast w przypadku baz *vehicle*, *ionosphere*, *vertebal*, *yeastME3*, *solar_flare* klasyfikator bagging zwiększył dokładność klasyfikacji średnio o 2-3%, czułość pozostała na niezmienionym poziomie, wzrosła natomiast specyficzność klasy mniejszościowej o 2-3% oraz miara G-mean. Należy zaznaczyć, że dokładność oraz pozostałe współczynniki rosły wraz ze zwiększeniem liczebności klasyfikatorów. Powyżej 50 klasyfikatorów przyrost był minimalny. Dokładność klasyfikatora baggging dla pozostałych baz została przedstawiona w tabeli 4.5. W tabelach 4.5 i 4.6 w kolumnach znajdują się klasyfikatory z różną maksymalną głębokością drzewa, a znak '-' oznacza drzewo bez ograniczenia.

Zbiór danych	Liczba est.	-	3	5	7	10	15	20
breast_cancer	-	0.63	**0.73**	0.72	0.67	0.66	0.63	0.63
	5	0.68	**0.7**	**0.7**	0.67	0.67	0.68	0.68
	10	**0.71**	**0.71**	0.7	0.67	**0.71**	**0.71**	**0.71**
	20	0.71	0.71	0.72	0.71	**0.74**	0.71	0.71
	50	0.7	**0.72**	**0.72**	0.7	0.71	0.71	0.7
cmc	-	0.68	**0.78**	0.76	0.71	0.72	0.68	0.68
	5	0.71	**0.78**	0.77	0.76	0.73	0.71	0.71
	10	0.73	**0.77**	**0.77**	0.76	0.74	0.73	0.73
	20	0.74	**0.78**	0.77	0.77	0.75	0.74	0.74
	50	0.75	**0.78**	0.77	0.77	0.76	0.74	0.75
hepatitis	-	0.66	0.66	**0.68**	0.66	0.66	0.66	0.66
	5	0.68	**0.7**	0.68	0.68	0.68	0.68	0.68
	10	0.75	0.75	**0.77**	0.75	0.75	0.75	0.75
	20	**0.72**	0.7	**0.72**	**0.72**	**0.72**	**0.72**	**0.72**
	50	**0.7**	0.69	**0.7**	**0.7**	**0.7**	**0.7**	**0.7**
haberman	-	0.66	**0.75**	**0.75**	0.73	0.63	0.66	0.66
	5	0.66	**0.74**	0.73	0.69	0.67	0.66	0.66
	10	0.66	0.72	**0.74**	0.72	0.65	0.66	0.66
	20	0.66	0.73	**0.74**	0.68	0.67	0.66	0.66
	50	0.68	**0.74**	0.73	0.71	0.69	0.68	0.68
glass	-	0.7	**0.82**	0.75	0.69	0.7	0.7	0.7
	5	0.73	**0.86**	0.78	0.73	0.73	0.73	0.73
	10	0.84	**0.86**	0.83	0.84	0.84	0.84	0.84
	20	0.86	**0.89**	0.87	0.86	0.86	0.86	0.86
	50	0.86	**0.88**	0.87	0.86	0.86	0.86	0.86
abalone16_29	-	0.91	**0.94**	0.93	0.93	0.91	0.91	0.91
	5	0.93	**0.94**	**0.94**	0.93	0.93	0.93	0.93
	10	0.93	**0.94**	**0.94**	0.93	0.93	0.93	0.93
	20	**0.94**	**0.94**	**0.94**	**0.94**	0.93	**0.94**	**0.94**
	50	0.93	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**
heart_cleveland	-	0.82	**0.86**	0.79	0.82	0.82	0.82	0.82
	5	0.84	**0.87**	0.84	0.84	0.84	0.84	0.84
	10	0.85	**0.86**	0.85	0.85	0.85	0.85	0.85
	20	0.84	**0.88**	0.85	0.84	0.84	0.84	0.84
	50	0.84	**0.87**	0.85	0.84	0.84	0.84	0.84
postoperative	-	0.67	**0.71**	0.7	0.69	0.7	0.67	0.67
	5	0.68	**0.72**	0.68	0.69	0.7	0.68	0.68
	10	0.7	**0.72**	0.7	0.69	0.7	0.7	0.7
	20	0.7	**0.71**	0.7	0.68	**0.71**	0.7	0.7
	50	0.66	**0.7**	0.69	0.67	0.64	0.66	0.66

Tablica 4.5: Dokładność klasyfikatora bagging drzewa decyzyjne dla parametrów: *max_features = 1* oraz *max_samples = 1*.

Zwiększając liczbę estymatorów wzrastała także dokładność, zwykle 2-3% w stosunku do pojedynczego drzewa decyzyjnego. Zdecydowanie najlepsze wyniki w tej grupie oraz w reszcie baz danych, uzyskały klasyfikatory z drzewem decyzyjnym z maksymalną głębokością równą 3. Wraz ze wzrostem dokładności poprawiła się także czułość klasy większościowej, średnio o 2-10%. Największy przyrost czułości nastąpił w klasyfikatorze bagging składającym się z 50 klasyfikatorów. Oprócz zbioru *breast_cancer*, w którym specyficzność klasy zdominowanej wzrosła o 5%, w pozostałych nastąpił wyraźny spadek rozpoznawalności klasy mniejszościowej. Zwiększając liczbę klasyfikatorów malał współczynnik specyficzności (tabela 4.6), a w przypadku bazy *glass* spadł do zera. Zauważono, że wzrost wykrywalności obu

klas nastąpił w bazach zawierających dużą liczbę przykładów bezpiecznych. W bazach trudniejszych do klasyfikacji, z dużą liczbą przykładów rzadkich oraz odstających, dokładność klasyfikacji klasy dominującej wzrosła kosztem wykrywalności klasy mniejszościowej.

Zbiór danych	Liczba est.	-	3	5	7	10	15	20
breast_cancer	-	0.38	0.31	0.32	0.31	**0.4**	0.38	0.38
	5	**0.39**	0.33	0.36	0.34	0.38	**0.39**	**0.39**
	10	0.35	0.32	0.36	0.33	**0.38**	0.35	0.35
	20	0.38	0.31	0.36	0.36	**0.45**	0.38	0.38
	50	**0.4**	0.31	0.34	0.36	**0.4**	0.39	**0.4**
cmc	-	0.36	**0.39**	0.29	0.26	0.35	0.38	0.36
	5	**0.3**	0.13	0.2	0.23	0.29	0.29	**0.3**
	10	**0.27**	0.14	0.21	0.22	**0.27**	0.26	**0.27**
	20	**0.29**	0.25	0.23	0.26	0.28	**0.29**	**0.29**
	50	**0.32**	0.23	0.22	0.26	0.3	0.31	0.31
hepatitis	-	**0.59**	0.5	0.56	0.56	**0.59**	**0.59**	**0.59**
	5	**0.56**	0.53	0.53	**0.56**	**0.56**	**0.56**	**0.56**
	10	0.47	**0.53**	0.47	0.47	0.47	0.47	0.47
	20	0.5	0.5	**0.53**	0.5	0.5	0.5	0.5
	50	0.5	0.5	**0.53**	0.5	0.5	0.5	0.5
haberman	-	0.26	0.32	0.22	0.2	**0.33**	0.26	0.26
	5	0.31	0.32	**0.38**	0.32	0.31	0.31	0.31
	10	0.27	0.16	**0.36**	0.3	0.33	0.27	0.27
	20	0.28	0.25	**0.36**	0.3	0.33	0.3	0.28
	50	0.36	0.27	0.3	0.32	**0.37**	0.36	0.36
glass	-	**0.18**	0.12	0.12	0.12	**0.18**	**0.18**	**0.18**
	5	**0.24**	0.0	**0.24**	**0.24**	**0.24**	**0.24**	**0.24**
	10	0.12	0.0	**0.18**	0.12	0.12	0.12	0.12
	20	0.0	0.0	**0.06**	0.0	0.0	0.0	0.0
	50	0.0	0.0	0.0	0.0	0.0	0.0	0.0
abalone16_29	-	**0.31**	0.09	0.11	0.23	0.28	**0.31**	**0.31**
	5	**0.21**	0.07	0.12	0.14	0.2	**0.21**	**0.21**
	10	0.17	0.06	0.13	0.14	**0.2**	0.18	0.17
	20	0.18	0.09	0.12	0.13	**0.2**	**0.2**	0.19
	50	0.17	0.08	0.1	0.15	**0.18**	**0.18**	**0.18**
heart_cleveland	-	**0.17**	0.03	**0.17**	**0.17**	**0.17**	**0.17**	**0.17**
	5	**0.2**	0.06	0.06	0.17	0.17	**0.2**	**0.2**
	10	**0.11**	0.03	0.09	**0.11**	**0.11**	**0.11**	**0.11**
	20	**0.09**	0.03	0.06	**0.09**	**0.09**	**0.09**	**0.09**
	50	**0.06**	0.03	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**
postoperative	-	0.17	0.08	0.12	**0.21**	0.17	0.17	0.17
	5	0.12	0.12	0.12	**0.17**	**0.17**	0.12	0.12
	10	**0.12**	**0.12**	0.08	0.08	**0.12**	**0.12**	**0.12**
	20	0.21	0.12	0.17	0.17	**0.25**	0.21	0.21
	50	**0.12**	0.04	**0.12**	**0.12**	**0.12**	**0.12**	**0.12**

Tablica 4.6: Specyficzność klasy mniejszościowej, dla klasyfikatora bagging z drzewem decyzyjnym, z ustawionymi parametrami: *max_features = 1* oraz *max_samples = 1*.

W czterech bazach *glass*, *abalone16_29*, *heart_cleveland*, *postoperative* wartość miary G-mean malała wraz ze zwiększeniem liczebności klasyfikatorów, zaś w pozostałych bazach wzrastała. Dla baz *transfusion*, *balance_scale* oraz *car* oprócz wzrostu jakości klasyfikacji klasy większościowej dla klasyfikatorów z drzewem decyzyjnym z maksymalną głębokością drzewa równą 3, nie stwierdzono różnic w pozostałych współczynnikach.

W kolejnym teście klasyfikatora bagging z drzewem decyzyjnym (plik *grid-search/bagging_tree.py*) obliczono najlepsze średnie ustawienia liczby atrybutów oraz liczby przykładów. Wyszukiwanie zachłanne wykonano dla parametrów *max_features=[0.4, 0.6, 0.7, 0.8, 0.9, 1.0]* oraz *max_samples=[0.4, 0.6, 0.7, 0.8, 0.9, 1.0]*. Test przeprowadzono dla drzewa bez ograniczenia maksymalnej głębokości oraz z ograniczeniem do poziomu 3, 5, 7, 10, 20. Klasyfikator bagging badano dla 5, 10, 15, 20, 50 i 100 klasyfikatorów. Dla każdego klasyfikatora wyłaniano najlepsze ustawienia, a następnie obliczono wartości średnie. Średnia liczba atrybutów wyniosła 0.85, średnia liczba przykładów 0.74, natomiast mediana wyniosła odpowiednio 0.9 oraz 0.8. Ponad połowa klasyfikatorów osiągnęła najlepszą klasyfikację ze wszystkimi atrybutami, a 20% klasyfikatorów ze wszystkimi przykładami. Kolejne obliczenia wykonano z parametrami *max_features = 0.9* oraz *max_samples = 0.8*.

Zbiór danych	Liczba est.	-	3	5	7	10	15	20
seeds	-	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**
	5	0.89	**0.9**	0.89	0.89	0.89	0.89	0.89
	10	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**
	20	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**
	50	**0.91**	0.9	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**
new_thyroid	-	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**
	5	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**
	10	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**
	20	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**
	50	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**
vehicle	-	0.93	0.9	0.93	**0.94**	**0.94**	0.93	0.93
	5	**0.95**	0.92	0.93	0.94	**0.95**	**0.95**	**0.95**
	10	**0.96**	0.93	**0.96**	**0.96**	**0.96**	**0.96**	**0.96**
	20	**0.96**	0.93	**0.96**	**0.96**	**0.96**	**0.96**	**0.96**
	50	**0.97**	0.94	0.95	0.96	**0.97**	**0.97**	**0.97**
ionosphere	-	0.86	0.86	**0.87**	**0.87**	0.86	0.86	0.86
	5	**0.9**	0.89	0.89	0.89	**0.9**	**0.9**	**0.9**
	10	**0.91**	0.9	0.9	**0.91**	**0.91**	**0.91**	**0.91**
	20	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**
	50	0.91	0.91	0.91	**0.92**	0.91	0.91	0.91
vertebal	-	0.71	0.71	0.72	**0.73**	0.71	0.71	0.71
	5	0.71	**0.72**	0.71	0.71	0.71	0.71	0.71
	10	0.71	0.71	0.71	**0.72**	0.71	0.71	0.71
	20	0.71	0.71	**0.72**	**0.72**	**0.72**	0.71	0.71
	50	**0.73**	0.72	**0.73**	**0.73**	**0.73**	**0.73**	**0.73**
yeastME3	-	0.93	**0.94**	**0.94**	**0.94**	0.93	0.93	0.93
	5	0.94	**0.95**	**0.95**	0.94	0.94	0.94	0.94
	10	0.94	**0.95**	**0.95**	0.94	0.94	0.94	0.94
	20	0.94	**0.95**	0.94	0.94	**0.95**	0.94	0.94
	50	**0.95**	**0.95**	0.94	0.94	**0.95**	**0.95**	**0.95**

Tablica 4.7: Dokładność klasyfikatora bagging drzewa decyzyjne, dla *max_features = 0.9* oraz *max_samples = 0.8*.

W tabeli 4.7 przedstawiono dokładność dla 6 zbiorów danych z największą liczbą przykładów bezpiecznych. Dokładność klasyfikacji poprawiła się głównie dla drzew z ograniczoną wysokością oraz z większa liczbą klasyfikatorów. Podobnie jak poprzednio nastąpił wzrost specyficzności klasy mniejszościowej, miary F-1 klasy

mniejszościowej (tabela 4.8) oraz miary G-mean. Natomiast wykrywalność klasy większościowej wzrosła minimalnie. W tabeli 4.9 przedstawiono dokładność dla pozostałych danych. Otrzymane wyniki różniły się w granicy błędu od wyników uzyskanych z domyślnymi parametrami. W stosunku do normalnego drzewa decyzyjnego nastąpił kilku procentowy wzrost dokładności. Meta-klasyfikator bagging najlepsze wyniki osiągał z drzewem decyzyjnym z maksymalną głębokością równą trzy. Zwiększając liczbę klasyfikatorów, rosła stoponiowo dokładność. Uzyskano podobne wartości specyficzności klasy mniejszościowej (tabela 4.10) oraz pozostałych współczynników jak z domyślnymi ustawieniami baggingu.

Zbiór danych	Liczba est.	-	3	5	7	10	15	20
seeds	-	**0.87**	**0.87**	**0.87**	**0.87**	**0.87**	**0.87**	**0.87**
	5	0.83	**0.85**	0.83	0.83	0.83	0.83	0.83
	10	**0.85**	**0.85**	**0.85**	**0.85**	**0.85**	**0.85**	**0.85**
	20	**0.85**	0.84	**0.85**	**0.85**	**0.85**	**0.85**	**0.85**
	50	**0.87**	0.86	**0.87**	**0.87**	**0.87**	**0.87**	**0.87**
new_thyroid	-	**0.88**	**0.88**	**0.88**	**0.88**	**0.88**	**0.88**	**0.88**
	5	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**
	10	**0.88**	**0.88**	**0.88**	**0.88**	**0.88**	**0.88**	**0.88**
	20	**0.88**	**0.88**	**0.88**	**0.88**	**0.88**	**0.88**	**0.88**
	50	**0.9**	0.88	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**
vehicle	-	0.86	0.81	0.84	**0.87**	**0.87**	0.86	0.86
	5	**0.88**	0.82	0.85	0.87	**0.88**	**0.88**	**0.88**
	10	**0.91**	0.85	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**
	20	**0.92**	0.85	0.91	0.91	**0.92**	**0.92**	**0.92**
	50	**0.93**	0.87	0.89	0.92	**0.93**	**0.93**	**0.93**
ionosphere	-	0.81	0.79	**0.82**	**0.82**	0.81	0.81	0.81
	5	**0.86**	0.84	0.83	0.84	**0.86**	**0.86**	**0.86**
	10	0.86	0.85	0.85	**0.87**	0.86	0.86	0.86
	20	0.87	0.87	**0.88**	0.87	0.87	0.87	0.87
	50	0.87	0.87	**0.88**	**0.88**	0.87	0.87	0.87
vertebal	-	0.63	0.62	0.64	**0.66**	0.63	0.63	0.63
	5	0.62	**0.63**	0.62	0.62	0.62	0.62	0.62
	10	0.61	0.61	**0.62**	**0.62**	0.61	0.61	0.61
	20	0.62	0.62	**0.63**	**0.63**	**0.63**	0.62	0.62
	50	**0.65**	0.63	0.64	**0.65**	**0.65**	**0.65**	**0.65**
yeastME3	-	0.68	**0.74**	0.72	0.72	0.69	0.68	0.68
	5	0.75	**0.78**	0.76	0.74	0.74	0.75	0.75
	10	0.72	**0.76**	0.74	0.73	0.72	0.73	0.72
	20	0.73	**0.76**	0.73	0.74	0.74	0.73	0.73
	50	0.74	0.74	0.74	0.73	0.74	**0.75**	0.74

Tablica 4.8: Miara F-1 klasy mniejszościowej. Klasyfikator bagging drzewa decyzyjne z parametrami *max_features = 0.9* oraz *max_samples = 0.8*.

Zbiór danych	Liczba est.	-	3	5	7	10	15	20
breast_cancer	-	0.63	**0.73**	**0.73**	0.71	0.66	0.63	0.63
	5	0.65	**0.74**	0.7	0.69	0.67	0.66	0.65
	10	0.71	**0.74**	0.72	0.71	0.71	0.71	0.71
	20	0.67	**0.71**	**0.71**	0.69	0.68	0.68	0.67
	50	0.69	**0.73**	0.7	0.71	0.69	0.69	0.69
cmc	-	0.68	**0.78**	0.76	0.71	0.71	0.68	0.68
	5	0.72	**0.78**	0.77	0.75	0.74	0.72	0.72
	10	0.74	**0.78**	**0.78**	0.76	0.74	0.74	0.74
	20	0.75	**0.78**	0.77	0.77	0.75	0.74	0.75
	50	0.74	0.77	**0.78**	0.77	0.76	0.75	0.75
hepatitis	-	0.7	0.66	**0.72**	0.7	0.7	0.7	0.7
	5	0.64	**0.74**	0.65	0.64	0.64	0.64	0.64
	10	0.71	**0.72**	**0.72**	0.7	0.71	0.71	0.71
	20	0.72	0.72	**0.73**	0.72	0.72	0.72	0.72
	50	**0.7**	0.68	0.69	**0.7**	**0.7**	**0.7**	**0.7**
haberman	-	0.66	**0.75**	**0.75**	0.72	0.67	0.64	0.66
	5	0.68	**0.75**	0.73	0.71	0.68	0.68	0.68
	10	0.7	**0.75**	**0.75**	0.73	0.71	0.7	0.7
	20	0.68	**0.75**	**0.75**	0.69	0.67	0.68	0.68
	50	0.67	**0.75**	0.74	0.71	0.67	0.67	0.67
glass	-	0.75	**0.82**	0.76	0.74	0.75	0.75	0.75
	5	0.86	**0.9**	0.86	0.86	0.86	0.86	0.86
	10	0.87	**0.9**	0.86	0.87	0.87	0.87	0.87
	20	0.85	**0.88**	0.8	0.85	0.85	0.85	0.85
	50	0.86	**0.89**	0.84	0.86	0.86	0.86	0.86
abalone16_29	-	0.91	**0.94**	0.93	0.93	0.91	0.91	0.91
	5	0.93	**0.94**	**0.94**	**0.94**	0.93	0.93	0.93
	10	0.93	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**	0.93
	20	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**
	50	0.93	**0.94**	**0.94**	**0.94**	0.93	0.93	0.93
heart_cleveland	-	0.8	**0.86**	0.8	0.8	0.8	0.8	0.8
	5	**0.85**	0.83	0.84	0.84	**0.85**	**0.85**	**0.85**
	10	**0.86**	0.85	0.84	**0.86**	**0.86**	**0.86**	**0.86**
	20	0.86	**0.87**	0.85	0.86	0.86	0.86	0.86
	50	0.84	**0.87**	0.85	0.84	0.84	0.84	0.84
postoperative	-	0.66	**0.73**	0.7	0.68	0.63	0.66	0.66
	5	0.64	**0.68**	0.62	0.64	0.64	0.64	0.64
	10	**0.67**	**0.67**	0.63	0.66	**0.67**	**0.67**	**0.67**
	20	0.64	**0.68**	0.66	0.64	0.64	0.64	0.64
	50	0.63	**0.68**	0.66	0.66	0.64	0.63	0.63

Tablica 4.9: Dokładność klasyfikatora bagging drzewa decyzyjne, dla *max_features = 0.9* oraz *max_samples = 0.8*.

Zbiór danych	Liczba est.	-	3	5	7	10	15	20
breast_cancer	-	**0.39**	0.31	0.34	0.32	**0.39**	**0.39**	**0.39**
	5	**0.38**	0.34	0.29	0.33	**0.38**	**0.38**	**0.38**
	10	**0.39**	**0.39**	0.32	0.35	**0.39**	**0.39**	**0.39**
	20	0.36	0.34	0.34	0.36	**0.38**	**0.38**	0.36
	50	**0.41**	0.32	0.32	0.35	0.4	**0.41**	**0.41**
cmc	-	0.36	**0.39**	0.29	0.26	0.33	**0.39**	0.36
	5	0.31	0.23	0.2	0.24	**0.32**	0.31	**0.32**
	10	0.28	0.15	0.2	0.22	**0.29**	**0.29**	0.28
	20	**0.3**	0.15	0.21	0.26	0.29	**0.3**	**0.3**
	50	**0.31**	0.19	0.23	0.24	0.28	0.3	0.3
hepatitis	-	**0.56**	0.5	0.53	0.53	**0.56**	**0.56**	**0.56**
	5	0.5	0.47	**0.53**	0.5	0.5	0.5	0.5
	10	0.47	0.5	**0.53**	0.47	0.47	0.47	0.47
	20	**0.59**	0.53	**0.59**	**0.59**	**0.59**	**0.59**	**0.59**
	50	**0.53**	0.5	0.5	**0.53**	**0.53**	**0.53**	**0.53**
haberman	-	0.28	**0.32**	0.22	0.2	0.28	0.31	0.28
	5	0.26	0.23	**0.32**	0.23	0.23	0.26	0.26
	10	0.25	0.23	0.25	**0.26**	0.2	0.25	0.25
	20	**0.3**	0.23	0.23	0.26	0.27	**0.3**	**0.3**
	50	**0.31**	0.27	0.3	0.26	0.28	**0.31**	**0.31**
glass	-	**0.24**	0.12	0.12	0.12	**0.24**	**0.24**	**0.24**
	5	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	10	**0.06**	0.0	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**
	20	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**
	50	0.0	0.0	0.0	0.0	0.0	0.0	0.0
abalone16_29	-	**0.33**	0.09	0.11	0.24	0.29	**0.33**	**0.33**
	5	**0.24**	0.08	0.09	0.18	0.21	**0.24**	**0.24**
	10	0.18	0.07	0.1	0.15	**0.23**	0.18	0.18
	20	**0.18**	0.07	0.09	0.11	**0.18**	**0.18**	**0.18**
	50	0.17	0.08	0.08	0.13	0.16	**0.18**	0.17
heart_cleveland	-	**0.17**	0.03	**0.17**	**0.17**	**0.17**	**0.17**	**0.17**
	5	**0.17**	0.11	0.09	0.11	**0.17**	**0.17**	**0.17**
	10	0.06	0.0	**0.11**	0.06	0.06	0.06	0.06
	20	0.06	0.0	**0.11**	0.09	0.06	0.06	0.06
	50	**0.09**	0.03	**0.09**	**0.09**	**0.09**	**0.09**	**0.09**
postoperative	-	**0.25**	0.08	0.08	0.17	**0.25**	**0.25**	**0.25**
	5	**0.12**	0.04	0.04	0.08	**0.12**	**0.12**	**0.12**
	10	**0.12**	0.0	0.08	0.08	**0.12**	**0.12**	**0.12**
	20	**0.08**	0.04	0.04	**0.08**	**0.08**	**0.08**	**0.08**
	50	**0.08**	0.04	0.04	**0.08**	**0.08**	**0.08**	**0.08**

Tablica 4.10: Specyficzność klasy mniejszościowej dla klasyfikatora bagging z drzewem decyzyjnym z ustawieniami *max_features = 0.9* oraz *max_samples = 0.8*.

4.1.3 Bagging z klasyfikatorem kNN

Ostatnim klasyfikatorem przetestowanym w metodzie bagging był klasyfikator k najbliższych sąsiadów. W podstawowym ustawieniu, klasyfikator kNN analizuje pięciu sąsiadów. W niniejszym teście postanowiono przetestować meta-klasyfikator bagging z klasyfikatorem kNN dla 1, 2, 3, 5 oraz 7 sąsiadów. Tak samo jak poprzednio, klasyfikatory bagging były budowane z 5, 10, 20 oraz 50 klasyfikatorów. Test ten przeprowadzono z pliku *bagging_knn.py*. W pierwszym etapie budowano modele bagging ze wszystkimi atrybutami oraz z maksymalną liczebnością zbiorów. Wyniki dokładności klasyfikacji zostały przedstawione w tabeli 4.11.

Zbiór danych	Liczba est.	1	2	3	5	7
breast_cancer	-	0.63	**0.68**	0.63	0.65	0.65
	5	0.59	0.59	0.57	0.6	**0.63**
	10	0.63	0.62	0.6	**0.64**	**0.64**
	20	0.63	0.62	0.64	0.64	**0.65**
	50	0.64	0.63	0.63	0.64	**0.65**
cmc	-	0.71	**0.76**	0.73	0.74	0.74
	5	0.72	0.74	0.73	**0.75**	**0.75**
	10	0.72	0.74	0.74	0.75	**0.76**
	20	0.71	0.74	0.74	0.75	**0.76**
	50	0.72	0.73	0.74	0.75	**0.76**
hepatitis	-	0.65	**0.77**	0.7	0.7	0.74
	5	0.63	0.71	0.69	0.71	**0.74**
	10	0.67	0.69	0.71	0.72	**0.74**
	20	0.68	0.7	0.7	0.72	**0.74**
	50	0.65	0.7	0.71	0.74	**0.75**
haberman	-	0.71	**0.72**	0.68	0.69	0.71
	5	0.69	0.7	**0.72**	**0.72**	**0.72**
	10	0.71	**0.72**	0.7	0.7	**0.72**
	20	**0.71**	**0.71**	0.7	0.69	**0.71**
	50	0.71	0.69	0.69	0.69	**0.72**
glass	-	0.78	0.88	0.84	0.88	**0.89**
	5	0.79	0.85	0.87	**0.91**	**0.91**
	10	0.79	0.82	0.86	**0.9**	**0.9**
	20	0.79	0.8	0.84	0.87	**0.89**
	50	0.78	0.84	0.85	0.87	**0.89**
abalone16_29	-	0.92	0.93	0.93	0.93	**0.94**
	5	0.92	**0.93**	**0.93**	**0.93**	**0.93**
	10	0.92	**0.93**	**0.93**	**0.93**	**0.93**
	20	0.92	0.93	0.93	**0.94**	**0.94**
	50	0.92	0.93	0.93	**0.94**	**0.94**
heart_cleveland	-	0.83	**0.88**	0.87	**0.88**	**0.88**
	5	0.83	0.86	**0.87**	**0.87**	**0.87**
	10	0.85	0.86	**0.88**	0.87	**0.88**
	20	0.84	0.87	**0.88**	**0.88**	**0.88**
	50	0.83	0.86	0.87	**0.88**	**0.88**
postoperative	-	0.63	**0.71**	0.67	0.7	0.69
	5	0.64	0.7	0.67	0.69	**0.72**
	10	0.68	0.67	0.68	**0.71**	**0.71**
	20	0.64	0.67	0.68	0.7	**0.71**
	50	0.66	0.68	0.67	**0.71**	0.7

Tablica 4.11: Dokładność klasyfikatora bagging z kNN, dla *max_features = 1.0* oraz *max_samples = 1.0*.

Metoda bagging nie poprawiła jakości klasyfikacji. Dla pojedynczego klasyfikatora kNN czy też grupy klasyfikatorów bagging wyniki są podobne i mieszczą się w granicy błędu. Specyficzność klasy mniejszościowej (tabela 4.12) również nie zwiększyła się, pozostała na takim samym poziomie niezależnie od liczby klasyfikatorów składowych. Zmieniając liczbę analizowanych sąsiadów można wpływać na czułość i specyficzność klas. Dla większej liczby sąsiadów np. 5 i 7, rosła skuteczność klasyfikacji oraz czułość klasy większościowej, jednocześnie spadała specyficzność klasy mniejszościowej. Klasyfikator kNN, z k większym od 10, nie wykrywał już przykładów klasy zdominowanej. Dla mniejszej liczby sąsiadów (2 i 3) specyficzność była większa. Największą skuteczność w klasyfikacji przykładów z klasy mniejszościowej osiągnięto przy analizowaniu tylko 1 sąsiada. Niestety, przy zastosowaniu

takich parametrów, wartość czułości uległa zmniejszeniu, a precyzja klasy mniejszościowej była na niskim poziomie (duża liczba błędnie zakwalifikowanych przykładów do klasy mniejszościowej).

Zbiór danych	Liczba est.	1	2	3	5	7
breast_cancer	-	**0.36**	0.08	0.28	0.2	0.18
	5	**0.33**	0.22	0.2	0.15	0.16
	10	**0.33**	0.27	0.25	0.19	0.16
	20	**0.33**	0.29	0.26	0.19	0.18
	50	**0.36**	0.25	0.22	0.14	0.14
cmc	-	**0.38**	0.17	0.29	0.28	0.24
	5	**0.35**	0.28	0.3	0.28	0.25
	10	**0.31**	0.29	0.29	0.27	0.24
	20	**0.32**	0.3	0.29	0.26	0.24
	50	**0.32**	0.29	0.28	0.27	0.25
hepatitis	-	**0.22**	0.03	0.16	0.06	0.0
	5	**0.19**	0.09	0.12	0.03	0.06
	10	**0.22**	0.12	0.09	0.09	0.03
	20	**0.22**	0.12	0.09	0.09	0.03
	50	**0.22**	0.12	0.09	0.06	0.0
haberman	-	**0.33**	0.1	0.32	0.25	0.22
	5	0.32	**0.35**	**0.35**	0.26	0.27
	10	**0.31**	0.26	**0.31**	0.23	0.27
	20	0.31	**0.32**	0.31	0.2	0.21
	50	**0.33**	0.3	0.3	0.22	0.23
glass	-	**0.29**	0.18	0.24	0.18	0.12
	5	**0.35**	0.24	0.29	0.18	0.0
	10	**0.29**	0.24	0.24	0.18	0.12
	20	**0.29**	0.18	0.24	0.18	0.12
	50	**0.29**	0.18	0.24	0.12	0.12
abalone16_29	-	**0.23**	0.08	0.18	0.13	0.1
	5	**0.23**	0.15	0.16	0.15	0.11
	10	**0.18**	0.14	0.15	0.1	0.09
	20	**0.2**	0.17	0.16	0.11	0.09
	50	**0.22**	0.18	0.16	0.11	0.09
heart_cleveland	-	**0.14**	0.0	0.0	0.0	0.0
	5	**0.11**	0.06	0.09	0.0	0.0
	10	**0.11**	0.09	0.06	0.0	0.0
	20	**0.14**	0.11	0.0	0.0	0.0
	50	**0.14**	0.03	0.0	0.0	0.0
postoperative	-	**0.21**	0.0	0.08	0.04	0.04
	5	**0.25**	0.12	0.04	0.04	0.0
	10	**0.12**	0.04	0.0	0.04	0.0
	20	**0.12**	0.0	0.0	0.04	0.0
	50	**0.17**	0.0	0.0	0.04	0.04

Tablica 4.12: Specyficzność klasy mniejszościowej dla klasyfikatora bagging z kNN i ustawieniami *max_features = 1.0* oraz *max_ samples = 1.0*.

W celu określenia najlepszej liczby atrybutów oraz wielkości zbiorów, podobnie jak poprzednio, wykonano wyszukiwanie zachłanne. Obliczenia przeprowadzono dla meta-klasyfikatora składającego się z 5, 10, 15, 20, 50 i 100 klasyfikatorów kNN z 1, 2, 3 i 5 sąsiadami. Najlepszą liczbę atrybutów szukano spośród [0.4, 0.6, 0.7, 0.8, 0.9, 1.0], a liczbę przykładów z [0.4, 0.6, 0.7, 0.8, 0.9, 1.0]. Dla każdego meta-klasyfikatora i zbioru danych wyłaniano najlepsze ustawienia. Przeprowadzony test znajduje się w pliku *gridsearch/bagging_knn.py*. Mediana liczby atrybutów wyniosła

1, a liczby przykładów 0.8. Natomiast średnia liczba atrybutów to 0.9, a średnia liczba przykładów to 0.78.

Badanie meta-klasyfikatora bagging z kNN powtórzono dla powyższych ustawień. Otrzymane wyniki były bardzo podobne jak podczas testu z ustawieniami domyślnymi, dokładność i jakość klasyfikacji nie zwiększyła się. Także w tym teście, zwiększenie liczby klasyfikatorów nie wpłynęło na poprawę wyników. Z powyższych powodów odstąpiono od prezentacji tych wyników.

4.2 Boosting

Do przetestowania metody boosting wybrano algorytm AdaBoost, autorstwa Yoava Freunda i Roberta Schapire, w zmodyfikowanej wersji znanej jako AdaBoost-SAMME.R. Algorytm wielokrotnie trenuje „słabe" klasyfikatory na tym samym zbiorze danych, w kolejnych iteracjach zwiększając wagę przykładów źle sklasyfikowanych. Zatem wybrany klasyfikator bazowy musi umożliwiać nadawanie prawdopodobieństwa lub wag przykładom. Jako klasyfikator bazowy wybrano drzewo decyzyjne oraz naiwny klasyfikator bayesowski. W klasyfikatorze kNN nie można nadawać wag przykładom, dlatego zrezygnowano z niego w badaniu. Budując klasyfikator AdaBoost należy wybrać liczbę iteracji (klasyfikatorów).

4.2.1 AdaBoost z naiwnym klasyfikatorem Bayesa

Test klasyfikatora AdaBoost wykonano w pliku *adaboost_NB.py*. Badanie przeprowadzono dla 5, 10, 15, 30, 50, 100 i 200 klasyfikatorów hierarchicznych. Zbudowany klasyfikator był stabilny, otrzymywane wyniki były powtarzalne. Zaobserwowano wzrost dokładności klasyfikacji (tabela 4.13) w połowie zbiorów danych dla 5, 10 i 15 klasyfikatorów. Dokładność klasyfikacji zwiększyła się średnio o 5%, dla bazy *glass* dokładność wzrosła dwa razy, a dla *yeastME3* trzy razy. Niestety, w pozostałych zbiorach zwiększył się błąd klasyfikacji. Zwiększenie liczby klasyfikatorów (powyżej 15) nie przełożyło się na poprawę wyników, a w większości przypadków dokładność klasyfikacji zmalała. W 16 bazach odnotowano wyraźny spadek specyficzności (rozpoznawalności) klasy mniejszościowej (tabela 4.14), a w pozostałych 6 zbiorach wystąpił kilkukrotny wzrost wykrywania klasy zdominowanej. Trzy z sześciu baz zawierały dużą liczbę obserwacji odstających. Potwierdzeniem wszystkich obserwacji są wyniki miary G-mean (tabela 4.15). Zazwyczaj wraz ze wzrostem wykrywalności jednej klasy, spada jakość klasyfikacji drugiej klasy. W AdaBoost z NKB zwiększyła się dokładność klasyfikacji klasy dominującej, kosztem klasy zdominowanej. Tylko w 6 zbiorach wystąpił wzrost miary G-mean, natomiast aż w 16 przypadkach osiągnięto gorsze wyniki niż w podstawowej wersji algorytmu NKB.

Zbiór danych	NKB	5	10	15	30	50	100	200
seeds	0.9	0.69	0.84	0.9	0.89	0.87	**0.91**	**0.91**
new_thyroid	0.96	0.76	0.94	**0.97**	0.96	**0.97**	0.94	0.94
vehicle	0.66	0.6	0.71	0.64	0.81	0.86	**0.87**	0.83
ionosphere	**0.87**	0.78	0.72	0.69	0.79	0.79	0.83	0.79
vertebal	**0.78**	0.73	0.75	0.68	0.67	0.6	0.72	0.72
yeastME3	0.27	0.47	**0.87**	0.46	0.75	0.84	0.84	0.84
ecoli	0.78	0.72	0.89	0.69	0.81	**0.9**	0.89	0.85
bupa	0.54	0.53	0.53	0.58	**0.6**	0.56	0.55	0.58
horse_colic	**0.78**	0.65	0.54	0.55	0.58	0.66	0.69	0.67
german	**0.73**	**0.73**	0.57	**0.73**	0.57	0.57	0.57	0.57
breast_cancer	**0.72**	0.63	0.36	0.53	0.35	0.35	0.35	0.35
cmc	0.68	0.66	**0.73**	0.67	0.63	0.63	0.63	0.63
hepatitis	0.66	0.63	0.46	**0.7**	0.61	0.55	0.45	0.6
haberman	**0.73**	0.67	0.55	0.48	0.6	0.67	0.69	0.72
transfusion	**0.74**	0.56	0.73	0.49	0.41	0.73	0.56	0.58
car	0.89	**0.95**	0.9	0.9	0.91	0.91	0.91	0.91
glass	0.48	**0.91**	0.78	0.66	0.74	0.79	0.87	0.88
abalone16_29	**0.68**	0.62	0.55	0.62	0.55	0.55	0.55	0.55
solar_flare	**0.65**	**0.65**	0.32	**0.65**	0.32	0.32	0.32	0.32
heart_cleveland	0.81	0.76	0.82	**0.88**	0.74	0.8	0.8	0.8
balance_scale	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**
postoperative	**0.67**	0.57	0.63	0.58	0.66	0.54	0.59	0.59

Tablica 4.13: Dokładność klasyfikatora AdaBoost z naiwnym klasyfikatorem Bayesa.

Zbiór danych	NKB	5	10	15	30	50	100	200
seeds	**0.91**	0.66	0.64	0.8	0.76	0.81	0.89	0.9
new_thyroid	0.87	0.9	0.67	0.83	0.77	**0.93**	0.6	0.6
vehicle	**0.84**	0.78	0.56	0.72	0.56	0.46	0.64	0.78
ionosphere	**0.76**	0.7	0.6	0.7	0.64	0.64	0.74	0.49
vertebal	0.87	0.8	0.61	0.59	0.79	0.79	**0.9**	0.82
yeastME3	**0.99**	0.88	0.29	0.66	0.52	0.49	0.49	0.49
ecoli	**0.94**	0.6	0.31	0.51	0.46	0.37	0.49	0.49
bupa	**0.74**	0.43	0.6	0.31	0.21	0.27	0.34	0.4
horse_colic	**0.75**	0.43	0.62	0.51	0.36	0.2	0.3	0.33
german	**0.62**	**0.62**	0.32	**0.62**	0.32	0.32	0.32	0.32
breast_cancer	0.44	0.49	**0.73**	0.66	**0.73**	0.71	0.71	0.71
cmc	**0.61**	0.5	0.03	0.5	0.28	0.28	0.28	0.28
hepatitis	**0.78**	0.47	0.44	0.56	0.12	0.38	0.25	0.16
haberman	0.17	0.23	0.35	**0.62**	0.3	0.22	0.26	0.23
transfusion	0.2	0.48	0.3	0.5	**0.58**	0.31	0.54	0.52
car	**1.0**	0.52	**1.0**	0.42	0.69	0.69	0.69	0.69
glass	**0.82**	0.0	0.18	0.18	0.0	0.12	0.06	0.06
abalone16_29	0.58	**0.61**	0.31	**0.61**	0.31	0.31	0.31	0.31
solar_flare	**0.93**	0.91	0.26	0.91	0.26	0.26	0.26	0.26
heart_cleveland	**0.63**	0.29	0.17	0.03	0.2	0.14	0.14	0.14
balance_scale	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
postoperative	0.17	**0.46**	0.25	0.33	0.29	0.42	0.42	0.42

Tablica 4.14: Specyficzność klasyfikatora AdaBoost z NKB.

Zbiór danych	NKB	5	10	15	30	50	100	200
seeds	**0.91**	0.68	0.78	0.88	0.85	0.86	**0.91**	**0.91**
new_thyroid	0.92	0.82	0.81	0.91	0.87	**0.96**	0.77	0.77
vehicle	0.72	0.65	0.65	0.67	0.7	0.67	0.78	**0.81**
ionosphere	**0.84**	0.76	0.68	0.69	0.75	0.75	0.81	0.68
vertebal	**0.8**	0.74	0.7	0.65	0.69	0.63	0.75	0.74
yeastME3	0.42	0.61	0.52	0.53	0.64	**0.66**	**0.66**	**0.66**
ecoli	**0.85**	0.66	0.55	0.6	0.62	0.6	0.67	0.66
bupa	**0.55**	0.51	0.53	0.49	0.42	0.45	0.49	0.54
horse_colic	**0.77**	0.58	0.55	0.54	0.51	0.43	0.53	0.54
german	**0.69**	**0.69**	0.46	**0.69**	0.46	0.46	0.46	0.46
breast_cancer	**0.6**	0.58	0.39	0.56	0.37	0.37	0.37	0.37
cmc	**0.65**	0.59	0.18	0.6	0.45	0.45	0.45	0.45
hepatitis	**0.7**	0.56	0.45	0.65	0.3	0.47	0.35	0.33
haberman	0.4	0.44	0.46	**0.52**	0.46	0.43	0.47	0.46
transfusion	0.43	0.53	0.51	0.49	0.45	0.52	0.55	**0.56**
car	0.94	0.71	**0.95**	0.62	0.8	0.8	0.8	0.8
glass	**0.61**	0.0	0.38	0.35	0.0	0.32	0.24	0.24
abalone16_29	**0.63**	0.61	0.42	0.61	0.42	0.42	0.42	0.42
solar_flare	**0.77**	0.76	0.29	0.76	0.29	0.29	0.29	0.29
heart_cleveland	**0.72**	0.48	0.39	0.17	0.4	0.36	0.36	0.36
balance_scale	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
postoperative	0.38	**0.53**	0.44	0.47	0.48	0.5	0.52	0.52

Tablica 4.15: Miara G-mean dla AdaBoost z NKB.

4.2.2 AdaBoost z drzewem decyzyjnym

Meta-klasyfikator AdaBoost z drzewem decyzyjnym przetestowano dla 5, 10, 20 i 50 klasyfikatorów. Użyto drzewo decyzyjne bez ograniczenia głębokości oraz drzewa z ograniczona głębokością do 3, 5, 7, 10, 15, 20 poziomu. Przeprowadzony test znajduje się w pliku *adaboost_tree.py*. Badanie przeprowadzono dla wszystkich danych. Klasyfikator AdaBoost nie poprawił wyników (utrzymały się na tym samym poziomie) zbiorów *seeds* i *vehicle* oraz poprawił tylko nieznacznie dokładność w zbiorach *vertebal*, *yeastME3*, *bupa*, *horse_colic*, *solar_flare*, *balance_scale*. Wyniki pozostałych danych zostały zaprezentowane w tabelach. W kolumnach umieszczono wyniki dla AdaBoost z różną liczbą klasyfikatorów (pauza oznacza pojedyncze drzewo decyzyjne), natomiast w wierszach znajduje się informacja o głębokości drzewa (pauza oznacza drzewo bez ograniczonej głębokości). W tabeli 4.16 znajdują się wyniki większości zbiorów danych. Meta-klasyfikator poprawił wyniki każdego zbioru przynajmniej minimalnie, a większości baz o kilka procent. Najlepsze wyniki osiągnął dla bazowych drzew decyzyjnych z głębokością 1 i 3 oraz dla większej liczby klasyfikatorów tj. 50 i 100. Rozpoznawalność klasy mniejszościowej (tabela specyficzności 4.17) również wzrosła o kilka procent, a klasyfikatory oparte o drzewa o głębokości 3 i 5 generowały zazwyczaj najlepsze wyniki.

Zbiór danych	Głębokość drzewa	-	5	10	20	50	100
vehicle	-	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**
	1	0.76	0.89	0.93	0.96	0.97	**0.98**
	3	0.9	0.94	0.94	0.96	**0.97**	**0.97**
	5	0.93	0.95	0.95	0.96	**0.97**	**0.97**
ionosphere	-	**0.88**	0.86	0.86	0.86	0.86	0.86
	1	0.79	0.88	0.87	**0.91**	0.89	**0.91**
	3	0.86	0.85	0.87	0.89	0.9	**0.92**
	5	0.87	0.88	0.89	0.91	**0.93**	**0.93**
ecoli	-	**0.88**	**0.88**	**0.88**	**0.88**	**0.88**	**0.88**
	1	0.78	**0.89**	**0.89**	0.88	0.85	0.87
	3	0.86	0.89	0.89	0.89	**0.9**	0.89
	5	0.88	0.88	0.88	**0.89**	0.88	0.88
german	-	**0.69**	**0.69**	**0.69**	**0.69**	**0.69**	**0.69**
	1	0.7	0.73	0.74	0.74	**0.76**	0.75
	3	0.74	0.74	**0.75**	0.74	0.73	0.73
	5	**0.74**	0.71	0.71	0.71	0.71	**0.74**
breast_cancer	-	0.63	0.69	0.69	**0.71**	0.69	0.69
	1	0.7	0.71	0.71	**0.72**	0.71	0.71
	3	**0.73**	0.7	0.71	0.69	0.66	0.65
	5	**0.73**	0.68	0.69	0.71	**0.73**	0.71
cmc	-	0.69	0.72	0.72	0.72	**0.73**	**0.73**
	1	0.77	0.77	**0.78**	0.76	0.77	0.76
	3	**0.78**	0.77	0.75	0.74	0.73	0.72
	5	**0.76**	0.74	0.71	0.72	0.72	0.72
hepatitis	-	0.66	**0.71**	**0.71**	**0.71**	**0.71**	**0.71**
	1	0.63	0.75	**0.77**	0.74	0.75	**0.77**
	3	0.66	0.78	0.79	0.81	**0.83**	**0.83**
	5	0.68	0.74	0.74	**0.77**	**0.77**	0.75
haberman	-	0.62	**0.66**	0.63	0.63	0.63	0.64
	1	0.74	**0.75**	**0.75**	0.73	0.72	0.7
	3	**0.75**	0.72	0.7	0.67	0.64	0.64
	5	**0.75**	0.69	0.68	0.68	0.69	0.68
transfusion	-	0.68	0.68	**0.69**	0.68	**0.69**	**0.69**
	1	**0.76**	**0.76**	**0.76**	**0.76**	**0.76**	0.74
	3	0.68	**0.74**	0.71	0.69	0.69	0.69
	5	0.67	0.69	**0.7**	0.69	**0.7**	**0.7**
car	-	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**
	1	0.75	0.82	0.88	**0.93**	0.83	0.83
	3	0.69	0.81	**0.95**	0.88	0.9	0.88
	5	0.67	**0.98**	0.86	0.94	0.86	0.86
glass	-	0.69	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**
	1	**0.92**	0.79	0.87	0.86	0.83	0.82
	3	0.82	0.79	0.8	0.81	0.83	**0.85**
	5	0.75	**0.8**	0.79	0.79	**0.8**	**0.8**
abalone16_29	-	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**
	1	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**
	3	**0.94**	0.93	0.93	0.92	0.92	0.92
	5	**0.93**	0.92	0.92	0.92	**0.93**	**0.93**
heart_cleveland	-	**0.82**	0.8	0.8	0.8	0.8	0.8
	1	**0.88**	0.86	0.85	0.84	0.83	0.84
	3	**0.86**	0.78	0.82	0.83	0.85	0.84
	5	0.81	0.82	**0.85**	0.84	0.84	0.84
postoperative	-	0.61	0.62	0.62	0.61	**0.63**	0.61
	1	**0.72**	0.69	0.69	0.68	0.67	0.67
	3	**0.71**	0.58	0.6	0.58	0.61	0.64
	5	**0.67**	0.59	0.59	0.59	0.66	0.64

Tablica 4.16: Dokładność klasyfikatora AdaBoost z drzewem decyzyjnym.

Zbiór danych	Głębokość drzewa	-	5	10	20	50	100
vehicle	-	**0.9**	0.88	0.88	0.88	0.88	0.88
	1	0.0	0.66	0.82	0.91	0.94	**0.95**
	3	0.93	0.86	0.87	0.92	0.93	**0.95**
	5	0.85	0.87	0.89	0.92	**0.94**	**0.94**
ionosphere	-	**0.84**	0.81	0.81	0.81	0.81	0.81
	1	0.49	**0.82**	0.76	0.8	0.79	0.81
	3	0.75	0.78	0.75	0.79	0.79	**0.81**
	5	0.79	0.79	0.82	0.82	**0.85**	**0.85**
ecoli	-	**0.69**	0.6	0.6	0.6	0.6	0.6
	1	**0.63**	0.6	0.51	0.4	0.43	0.43
	3	0.49	0.54	0.6	0.51	**0.63**	0.6
	5	**0.63**	0.54	0.54	0.54	0.51	0.49
german	-	**0.48**	0.46	0.46	0.46	0.46	0.46
	1	0.0	0.42	0.42	0.45	**0.51**	0.5
	3	0.42	0.51	**0.54**	0.52	0.52	0.51
	5	0.4	0.46	**0.47**	**0.47**	0.45	0.45
breast_cancer	-	**0.4**	0.38	0.33	0.33	0.31	0.29
	1	**0.44**	0.41	0.35	0.38	0.39	0.38
	3	0.31	0.38	0.4	**0.44**	0.41	0.36
	5	0.33	0.36	0.35	**0.41**	0.4	0.35
cmc	-	**0.37**	0.33	0.33	0.32	0.31	0.31
	1	0.0	0.08	0.19	0.21	0.21	**0.22**
	3	**0.39**	0.32	0.3	0.3	0.36	0.35
	5	0.29	0.34	0.31	0.34	**0.35**	0.33
hepatitis	-	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**
	1	0.53	0.41	**0.56**	0.47	0.53	**0.56**
	3	0.5	0.5	0.56	0.56	0.62	**0.66**
	5	0.56	0.56	0.56	**0.62**	0.56	0.56
haberman	-	0.31	0.25	0.42	0.42	0.42	**0.46**
	1	0.0	0.23	**0.25**	0.2	0.22	**0.25**
	3	0.32	0.32	0.22	0.37	0.42	**0.44**
	5	0.22	0.33	0.36	**0.37**	0.28	0.31
transfusion	-	0.29	0.27	**0.32**	0.3	0.28	0.28
	1	0.0	0.39	**0.4**	**0.4**	**0.4**	0.36
	3	0.45	**0.46**	0.36	0.29	0.3	0.28
	5	**0.36**	0.32	0.3	0.31	0.32	0.3
car	-	**0.46**	**0.46**	**0.46**	**0.46**	**0.46**	**0.46**
	1	0.32	0.46	0.49	**0.54**	0.51	0.49
	3	0.32	0.52	**0.68**	0.58	0.65	0.6
	5	0.46	**0.6**	0.57	0.55	0.58	**0.6**
glass	-	**0.24**	0.18	0.18	0.18	0.18	0.18
	1	0.0	**0.24**	**0.24**	0.12	0.06	0.18
	3	0.12	0.06	0.06	**0.24**	0.18	0.18
	5	0.12	0.12	0.18	**0.24**	**0.24**	0.18
abalone16_29	-	**0.33**	0.31	0.31	0.31	0.31	0.31
	1	0.0	0.03	0.11	0.16	0.18	**0.2**
	3	0.09	0.21	**0.26**	0.24	0.24	0.22
	5	0.11	**0.25**	0.22	0.23	0.2	0.15
heart_cleveland	-	**0.2**	0.17	0.17	0.17	0.17	0.17
	1	0.0	0.17	0.17	**0.2**	0.17	**0.2**
	3	0.03	**0.09**	0.03	0.0	0.06	0.0
	5	**0.2**	0.06	0.09	0.03	0.03	0.03
postoperative	-	**0.17**	0.12	**0.17**	0.12	0.12	0.12
	1	0.0	**0.12**	0.08	0.08	**0.12**	**0.12**
	3	0.08	0.17	0.17	**0.21**	**0.21**	0.17
	5	0.08	**0.25**	0.12	0.17	0.17	0.17

Tablica 4.17: Specyficzność klasyfikatora AdaBoost z drzewem decyzyjnym.

4.2.3 Stacking

Do zbudowania meta-klasyfikatora stacking wykorzystano klasyfikator kNN, drzewo decyzyjne oraz naiwny klasyfikator Bayesa. Wszystkie klasyfikatory bazowe były tworzone z ustawieniami domyślnymi, czyli klasyfikator kNN analizował 5 sąsiadów, a drzewo decyzyjne było bez ograniczenia głębokości. W badaniach jako końcowy meta-klasyfikator wykorzystywano regresję logistyczną, pojedynczą sieć neuronową oraz wielowarstwową sieć neuronową (MLP). Najlepsze wyniki uzyskano z wielowarstwową siecią neuronową i to z nią zaprezentowano wynik działania meta-klasyfikatora. Każdy klasyfikator bazowy trenowany był osobno, a następnie na otrzymanych wynikach trenowana była sieć neuronowa. Klasyfikację danych rozpoczynają klasyfikatory bazowe, a o końcowej klasie decyduje meta-klasyfikator. W drugiej wersji meta-klasyfikatora, modele bazowe, zamiast propozycji klas, generują prawdopodobieństwo klas, a następnie dane te wykorzystywane są jako wejście dla głównego klasyfikatora. Skrypt testowy znajduje się w pliku *stacking_ cmp.py*. W tabeli 4.18 przedstawiono dokładność klasyfikatora stacking. „STK" oznacza wersję z klasami jako atrybuty wejścia dla meta-klasyfikatora, natomiast „STK PROBA" to wersja z prawdopodobieństwem klas.

	KNN	TREE	NKB	STK	STK PROBA	VOTING
seeds	**0.92**	0.9	0.9	0.9	0.9	**0.92**
new_thyroid	0.96	**0.97**	0.96	**0.97**	**0.97**	**0.97**
vehicle	0.92	**0.94**	0.66	**0.94**	**0.94**	**0.94**
ionosphere	0.82	0.89	0.87	0.89	0.89	**0.92**
vertebal	0.74	0.72	**0.78**	0.72	0.72	0.74
yeastME3	**0.95**	0.93	0.27	0.93	0.93	0.94
ecoli	0.89	0.88	0.78	0.88	**0.9**	**0.9**
bupa	**0.68**	0.64	0.54	0.64	0.64	0.67
horse_colic	0.71	**0.81**	0.78	**0.81**	**0.81**	0.8
german	0.69	0.68	**0.73**	0.68	0.68	**0.73**
breast_cancer	0.65	0.63	**0.72**	0.63	0.62	0.69
cmc	**0.74**	0.68	0.68	0.68	0.68	0.73
hepatitis	0.7	0.66	0.66	0.66	0.66	**0.71**
haberman	0.69	0.68	**0.73**	0.68	0.68	0.68
transfusion	0.68	0.69	**0.74**	0.69	0.69	**0.74**
car	**0.92**	0.67	0.89	0.89	0.9	0.89
glass	**0.88**	0.68	0.48	0.68	0.68	0.82
abalone16_29	**0.93**	0.91	0.68	0.91	0.91	0.92
solar_flare	**0.95**	0.94	0.65	0.94	0.92	0.94
heart_cleveland	**0.88**	0.81	0.81	0.81	0.81	0.86
balance_scale	**0.92**	0.85	**0.92**	0.85	0.85	**0.92**
postoperative	0.7	0.64	0.67	0.64	0.62	**0.71**

Tablica 4.18: Dokładność klasyfikatora stacking.

Meta-klasyfikator stacking tylko w 4 przypadkach osiągnął wartości równe klasyfikatorowi bazowemu. W pozostałych bazach, osiągał zazwyczaj lepszą dokładność niż najgorszy klasyfikator, ale niższą niż najlepszy. W 7 bazach zwiększył rozpoznawalność klasy mniejszościowej (tabela 4.19), a w pozostałych osiągnął wartości lepsze niż najgorszy klasyfikator. Potwierdzeniem powyższego są wyniki miary G-mean

(tabela 4.20). Dla porównania, w ostatniej tabeli zamieszczono wyniki głosowania większościowego (te same trzy klasyfikatory bazowe). Lepszą dokładność osiągnięto głosując, natomiast klasyfikator stacking lepiej rozpoznawał klasę mniejszościową.

	KNN	TREE	NKB	STK	STK PROBA	VOTING
seeds	**0.91**	0.84	**0.91**	0.84	0.84	0.9
new_thyroid	0.73	**0.87**	**0.87**	**0.87**	**0.87**	0.8
vehicle	0.84	0.9	0.84	0.9	0.9	**0.95**
ionosphere	0.55	**0.87**	0.76	**0.87**	**0.87**	0.79
vertebal	0.79	0.77	**0.87**	0.77	0.77	0.81
yeastME3	0.68	0.72	**0.99**	0.72	0.72	0.8
ecoli	0.54	0.63	**0.94**	0.63	0.63	0.77
bupa	0.48	0.57	**0.74**	0.57	0.57	0.61
horse_colic	0.54	**0.77**	0.75	**0.77**	**0.77**	0.74
german	0.32	0.49	**0.62**	0.49	0.49	0.46
breast_cancer	0.2	0.4	**0.44**	0.4	0.4	0.31
cmc	0.28	0.36	**0.61**	0.36	0.38	0.36
hepatitis	0.06	0.62	**0.78**	0.62	0.62	0.56
haberman	0.25	**0.3**	0.17	**0.3**	**0.3**	0.17
transfusion	**0.31**	0.3	0.2	0.3	**0.31**	0.26
car	0.43	0.46	**1.0**	0.46	0.46	0.48
glass	0.18	0.24	**0.82**	0.24	0.24	0.24
abalone16_29	0.13	0.33	**0.58**	0.33	0.33	0.27
solar_flare	0.05	0.09	**0.93**	0.14	0.12	0.14
heart_cleveland	0.0	0.23	**0.63**	0.23	0.23	0.17
balance_scale	0.0	**0.02**	0.0	**0.02**	**0.02**	0.0
postoperative	0.04	0.17	0.17	0.17	**0.21**	0.12

Tablica 4.19: Specyficzność klasy mniejszościowej klasyfikatora stacking.

	KNN	TREE	NKB	STK	STK PROBA	VOTING
seeds	**0.92**	0.88	0.91	0.88	0.88	0.91
new_thyroid	0.86	**0.92**	**0.92**	**0.92**	**0.92**	0.89
vehicle	0.89	0.93	0.72	0.93	0.93	**0.94**
ionosphere	0.73	**0.88**	0.84	**0.88**	**0.88**	**0.88**
vertebal	0.75	0.73	**0.8**	0.73	0.73	0.76
yeastME3	0.82	0.83	0.42	0.83	0.83	**0.87**
ecoli	0.71	0.76	**0.85**	0.76	0.76	0.84
bupa	0.63	0.63	0.55	0.63	0.63	**0.66**
horse_colic	0.67	**0.8**	0.77	**0.8**	**0.8**	0.79
german	0.52	0.61	**0.69**	0.61	0.61	0.63
breast_cancer	0.41	0.54	**0.6**	0.54	0.54	0.51
cmc	0.49	0.53	**0.65**	0.53	0.54	0.55
hepatitis	0.23	0.65	**0.7**	0.65	0.65	0.65
haberman	0.46	**0.49**	0.4	**0.49**	**0.49**	0.39
transfusion	**0.5**	0.49	0.43	0.49	**0.5**	0.48
car	0.63	0.56	**0.94**	0.65	0.65	0.66
glass	0.41	0.41	**0.61**	0.41	0.41	0.45
abalone16_29	0.35	0.56	**0.63**	0.56	0.56	0.51
solar_flare	0.21	0.3	**0.77**	0.37	0.33	0.37
heart_cleveland	0.0	0.45	**0.72**	0.45	0.45	0.4
balance_scale	0.0	**0.14**	0.0	**0.14**	**0.14**	0.0
postoperative	0.2	0.37	0.38	0.37	**0.4**	0.34

Tablica 4.20: G-mean klasyfikatora stacking.

4.2.4 Porównanie meta-metod

W ostatnim etapie testów meta-metod wykonano zbiorcze porównanie meta-klasyfikatorów. Do porównania wybrano bagging z naiwnym klasyfikatorem Bayesa (bag NKB), bagging z drzewem decyzyjnym (Bag Tree), bagging z klasyfikatorem kNN (Bag kNN), AdaBoost z naiwnym klasyfikatorem Bayesa (AB NKB), AdaBoost z drzewem decyzyjnym (AB TREE), las losowy (RF) oraz stacking z naiwnym klasyfikatorem Bayesa, drzewem decyzyjnym i klasyfikatorem kNN. Pierwszy test wykonano dla 50 klasyfikatorów (iteracji) oraz użyto klasyfikatorów z ustawieniami domyślnymi. Otrzymane wyniki dokładności przedstawiono w tabeli 4.21. Zdecydowanie najczęściej najlepszą dokładność osiągał las losowy. Klasyfikator bagging kNN osiągnął wysoką dokładność w zbiorach zawierających dużo przykładów rzadkich i odstających. Wysoka skuteczność obu klasyfikatorów odbiła się na niskiej wykrywalności klasy mniejszościowej (tabela specyficzności 4.22) przez las losowy i prawie zerowej w przypadku klasyfikatora bagging kNN. Najlepszą wykrywalność klasy mniejszościowej oraz obu klas (miara G-mean tabela 4.23) osiągnął klasyfikator bagging z naiwnym klasyfikatorem Bayesa. Większość otrzymanych wyników różniła się tylko o kilka procent. Jednakże miały miejsce przypadki, że jeden klasyfikator osiągał zdecydowanie gorsze wyniki dla niektórych baz, np. bagging NKB miał słabą skuteczność klasyfikacji dla baz: *vehicle*, *yeastME3*, *glass*, a AdaBoost NKB dla baz *breast_ cancer* i *solar_ flare*.

Zbiór danych	Bag NKB	Bag TREE	Bag kNN	AB NKB	AB Tree	RF	Stacking
seeds	0.9	0.9	**0.92**	0.87	0.91	0.9	0.9
new_thyroid	0.96	**0.97**	0.96	**0.97**	**0.97**	**0.97**	**0.97**
vehicle	0.67	0.96	0.93	0.86	0.94	**0.97**	0.94
ionosphere	0.87	0.9	0.81	0.79	0.86	**0.93**	0.89
vertebal	**0.77**	0.72	0.73	0.6	0.73	0.72	0.72
yeastME3	0.24	0.94	**0.95**	0.84	0.92	**0.95**	0.93
ecoli	0.8	0.88	0.89	0.9	0.88	**0.92**	0.88
bupa	0.57	0.7	0.68	0.56	0.64	**0.72**	0.65
horse_colic	0.77	0.85	0.73	0.66	0.8	**0.86**	0.81
german	0.71	0.76	0.69	0.57	0.68	**0.77**	0.68
breast_cancer	0.72	0.67	0.65	0.35	**0.73**	0.71	0.65
cmc	0.68	0.74	0.75	0.63	0.73	**0.76**	0.69
hepatitis	0.68	0.72	0.71	0.55	0.66	**0.83**	0.65
haberman	**0.74**	0.66	0.69	0.67	0.63	0.7	0.67
transfusion	**0.74**	0.69	0.72	0.73	0.69	0.7	0.68
car	0.9	0.68	**0.94**	0.91	0.67	0.92	0.89
glass	0.53	0.88	0.87	0.79	0.75	**0.89**	0.71
abalone16_29	0.68	**0.94**	**0.94**	0.55	0.91	**0.94**	0.91
solar_flare	0.63	0.93	**0.95**	0.32	0.94	0.94	0.93
heart_cleveland	0.81	0.84	**0.88**	0.8	0.79	0.87	0.81
balance_scale	**0.92**	0.87	**0.92**	**0.92**	0.85	0.89	0.85
postoperative	0.66	0.66	**0.71**	0.54	0.6	0.68	0.64

Tablica 4.21: Dokładność - porównanie meta-klasyfikatorów.

Zbiór danych	Bag NKB	Bag TREE	Bag kNN	AB NKB	AB Tree	RF	Stacking
seeds	**0.91**	0.84	**0.91**	0.81	0.87	0.84	0.84
new_thyroid	0.87	0.87	0.73	**0.93**	0.87	0.87	0.87
vehicle	0.84	0.92	0.86	0.46	0.88	**0.94**	0.88
ionosphere	0.76	0.83	0.5	0.64	0.82	0.84	**0.87**
vertebal	**0.86**	0.77	0.78	0.79	0.77	0.75	0.78
yeastME3	**0.99**	0.72	0.68	0.49	0.71	0.7	0.71
ecoli	**0.94**	0.54	0.57	0.37	0.6	0.54	0.63
bupa	**0.72**	0.48	0.48	0.27	0.57	0.52	0.59
horse_colic	0.74	0.76	0.57	0.2	0.76	0.76	**0.78**
german	**0.68**	0.47	0.29	0.32	0.48	0.4	0.44
breast_cancer	0.44	0.4	0.15	**0.71**	0.39	0.33	0.42
cmc	**0.61**	0.29	0.26	0.28	0.31	0.29	0.38
hepatitis	**0.75**	0.53	0.0	0.38	0.56	0.59	0.56
haberman	0.21	0.31	0.25	0.22	**0.43**	0.27	0.28
transfusion	0.21	**0.31**	0.27	**0.31**	0.28	**0.31**	0.3
car	**1.0**	0.46	0.45	0.69	0.46	0.46	0.46
glass	**0.76**	0.0	0.12	0.12	0.18	0.06	0.18
abalone16_29	**0.57**	0.17	0.1	0.31	0.31	0.1	0.31
solar_flare	**0.93**	0.14	0.05	0.26	0.07	0.16	0.16
heart_cleveland	**0.57**	0.09	0.0	0.14	0.14	0.03	0.11
balance_scale	0.0	0.0	0.0	0.0	**0.04**	0.0	0.02
postoperative	0.17	0.12	0.0	**0.42**	0.17	0.04	0.17

Tablica 4.22: Specyficzność klasy mniejszościowej - porównanie meta-klasyfikatorów.

Zbiór danych	Bag NKB	Bag TREE	Bag kNN	AB NKB	AB Tree	RF	Stacking
seeds	0.91	0.88	**0.92**	0.86	0.9	0.88	0.88
new_thyroid	0.92	0.92	0.86	**0.96**	0.92	0.93	0.92
vehicle	0.72	0.95	0.9	0.67	0.92	**0.96**	0.92
ionosphere	0.84	0.88	0.7	0.75	0.85	**0.9**	0.88
vertebal	**0.79**	0.73	0.74	0.63	0.74	0.72	0.73
yeastME3	0.38	**0.83**	0.82	0.66	0.82	**0.83**	0.82
ecoli	**0.86**	0.71	0.73	0.6	0.74	0.72	0.76
bupa	0.58	0.64	0.63	0.45	0.63	**0.67**	0.64
horse_colic	0.76	0.83	0.68	0.43	0.79	**0.84**	0.8
german	**0.7**	0.64	0.5	0.46	0.61	0.61	0.59
breast_cancer	**0.6**	0.56	0.36	0.37	0.58	0.54	0.56
cmc	**0.65**	0.5	0.48	0.45	0.52	0.51	0.54
hepatitis	0.71	0.64	0.0	0.47	0.62	**0.73**	0.62
haberman	0.44	0.49	0.46	0.43	**0.55**	0.48	0.48
transfusion	0.43	0.5	0.48	**0.52**	0.48	0.51	0.49
car	**0.94**	0.56	0.65	0.8	0.56	0.66	0.65
glass	**0.62**	0.0	0.33	0.32	0.38	0.24	0.37
abalone16_29	**0.63**	0.41	0.31	0.42	0.54	0.31	0.55
solar_flare	**0.76**	0.37	0.21	0.29	0.26	0.4	0.4
heart_cleveland	**0.69**	0.28	0.0	0.36	0.35	0.17	0.32
balance_scale	0.0	0.0	0.0	0.0	**0.19**	0.0	0.14
postoperative	0.37	0.33	0.0	**0.5**	0.36	0.19	0.37

Tablica 4.23: G-mean - porównanie meta-klasyfikatorów.

Drugi test porównujący wykonano dla 100 klasyfikatorów bazowych oraz ogra-

niczono głębokość drzewa do 3 poziomu. Ponownie najlepszym klasyfikatorem pod względem dokładności (tabela 4.24) okazał się las losowy, a klasyfikator bagging NKB najlepiej wykrywał klasę mniejszościową (tabela 4.25).

Zbiór danych	Bag NKB	Bag TREE	Bag kNN	AB NKB	AB Tree	RF	Stacking
seeds	0.9	0.91	**0.94**	0.91	0.91	0.91	0.91
new_thyroid	0.96	**0.97**	0.96	0.94	**0.97**	**0.97**	**0.97**
vehicle	0.66	**0.97**	0.92	0.87	0.94	**0.97**	0.95
ionosphere	0.87	0.9	0.82	0.83	0.86	**0.92**	0.88
vertebal	**0.78**	0.72	0.73	0.72	0.72	0.72	0.72
yeastME3	0.24	0.94	**0.95**	0.84	0.93	**0.95**	0.93
ecoli	0.79	0.9	0.89	0.89	0.88	**0.91**	0.88
bupa	0.55	0.71	0.68	0.55	0.62	**0.72**	0.65
horse_colic	0.78	**0.86**	0.71	0.69	0.8	0.85	0.78
german	0.72	0.75	0.7	0.57	0.69	**0.77**	0.7
breast_cancer	0.72	0.69	0.66	0.35	0.7	**0.73**	0.64
cmc	0.68	0.74	0.75	0.63	0.73	**0.76**	0.69
hepatitis	0.67	0.72	0.7	0.45	0.68	**0.81**	0.7
haberman	**0.74**	0.69	0.69	0.69	0.64	0.72	0.65
transfusion	**0.74**	0.68	0.73	0.56	0.68	0.7	0.69
car	0.9	0.68	**0.94**	0.91	0.67	0.87	0.89
glass	0.51	**0.9**	0.87	0.87	0.68	0.88	0.68
abalone16_29	0.68	**0.94**	**0.94**	0.55	0.91	**0.94**	0.91
solar_flare	0.61	0.94	**0.95**	0.32	0.93	0.94	0.94
heart_cleveland	0.8	0.84	**0.88**	0.8	0.8	0.85	0.82
balance_scale	**0.92**	0.87	**0.92**	**0.92**	0.85	0.89	0.85
postoperative	0.63	0.67	**0.68**	0.59	0.58	0.66	0.66

Tablica 4.24: Dokładność - porównanie meta-klasyfikatorów dla testu nr 2.

Zwiększenie liczby klasyfikatorów tylko w niektórych przypadkach pozwoliło na zwiększenie skuteczności klasyfikacji, większość klasyfikatorów osiągnęła podobny poziom skuteczności. Zwiększając liczbę klasyfikatorów w meta-klasyfikatorze bagging NKB uzyskano dużo lepszą wykrywalność klasy mniejszościowej w bazie *vehicle*. Jednak zwiększona liczba klasyfikatorów wpłynęła negatywnie na specyficzność klasy mniejszościowej dla bazy *new_thyroid*. Zmieniając ustawienia klasyfikatora można poprawić jakość klasyfikacji dla niektórych zbiorów danych, a pogorszyć dla innych. Aby uzyskać możliwe najlepszą klasyfikację, należy dla każdego zbioru osobno dobierać klasyfikator i ustawienia. Najbardziej stabilne wyniki (dokładność klasyfikacji oraz specyficzność klasy mniejszościowej) niezależnie od zbioru danych, zazwyczaj minimalnie gorsze od najlepszego klasyfikatora, otrzymano z meta-klasyfikatora stacking. Był to najbardziej uniwersalny klasyfikator.

Zbiór danych	Bag NKB	Bag TREE	Bag kNN	AB NKB	AB Tree	RF	Stacking
seeds	0.91	0.87	**0.96**	0.89	0.87	0.89	0.86
new_thyroid	**0.87**	**0.87**	0.73	0.6	**0.87**	**0.87**	**0.87**
vehicle	0.84	0.92	0.84	0.64	0.89	**0.94**	0.88
ionosphere	0.76	0.83	0.52	0.74	0.81	**0.86**	0.85
vertebal	0.87	0.76	0.76	**0.9**	0.78	0.75	0.76
yeastME3	**0.99**	0.71	0.68	0.49	0.72	0.69	0.69
ecoli	**0.94**	0.63	0.54	0.49	0.66	0.51	0.6
bupa	**0.74**	0.52	0.47	0.34	0.53	0.5	0.57
horse_colic	0.75	**0.76**	0.55	0.3	**0.76**	0.74	0.74
german	**0.68**	0.46	0.3	0.32	0.47	0.4	0.48
breast_cancer	0.44	0.39	0.19	**0.71**	0.34	0.39	0.38
cmc	**0.6**	0.29	0.27	0.28	0.32	0.27	0.39
hepatitis	**0.72**	0.5	0.03	0.25	0.59	0.53	0.59
haberman	0.19	0.31	0.23	0.26	**0.53**	0.3	0.28
transfusion	0.21	0.31	0.28	**0.54**	0.28	0.32	0.31
car	**1.0**	0.46	0.43	0.69	0.46	0.45	0.46
glass	**0.82**	0.0	0.12	0.06	0.18	0.06	0.18
abalone16_29	**0.58**	0.18	0.11	0.31	0.32	0.11	0.3
solar_flare	**0.93**	0.14	0.05	0.26	0.05	0.09	0.14
heart_cleveland	**0.57**	0.06	0.0	0.14	0.17	0.03	0.2
balance_scale	0.0	0.0	0.0	0.0	0.02	0.0	**0.04**
postoperative	0.17	0.12	0.0	**0.42**	0.17	0.08	0.17

Tablica 4.25: Specyficzność klasy mniejszościowej - porównanie meta-klasyfikatorów dla testu nr 2.

Zbiór danych	Bag NKB	Bag TREE	Bag kNN	AB NKB	AB Tree	RF	Stacking
seeds	0.91	0.9	**0.94**	0.91	0.9	0.91	0.9
new_thyroid	0.92	0.92	0.86	0.77	0.92	**0.93**	0.92
vehicle	0.72	0.95	0.9	0.78	0.92	**0.96**	0.92
ionosphere	0.84	0.88	0.72	0.81	0.85	**0.91**	0.87
vertebal	**0.8**	0.73	0.74	0.75	0.74	0.73	0.73
yeastME3	0.38	**0.83**	0.82	0.66	**0.83**	0.82	0.81
ecoli	**0.86**	0.76	0.71	0.67	0.77	0.7	0.74
bupa	0.55	**0.66**	0.62	0.49	0.61	**0.66**	0.63
horse_colic	0.77	**0.83**	0.67	0.53	0.79	0.82	0.77
german	**0.71**	0.63	0.51	0.46	0.61	0.61	0.61
breast_cancer	**0.6**	0.56	0.4	0.37	0.54	0.58	0.53
cmc	**0.65**	0.5	0.49	0.45	0.52	0.5	0.55
hepatitis	**0.69**	0.62	0.17	0.35	0.65	**0.69**	0.66
haberman	0.42	0.51	0.45	0.47	**0.6**	0.51	0.47
transfusion	0.43	0.5	0.5	**0.55**	0.48	0.52	0.5
car	**0.94**	0.56	0.64	0.8	0.56	0.63	0.65
glass	**0.63**	0.0	0.33	0.24	0.36	0.24	0.36
abalone16_29	**0.63**	0.42	0.33	0.42	0.55	0.33	0.53
solar_flare	**0.74**	0.37	0.21	0.29	0.21	0.3	0.37
heart_cleveland	**0.69**	0.23	0.0	0.36	0.39	0.17	0.42
balance_scale	0.0	0.0	0.0	0.0	0.14	0.0	**0.19**
postoperative	0.37	0.33	0.0	**0.52**	0.35	0.27	0.37

Tablica 4.26: G-mean - porównanie meta-klasyfikatorów, badania nr 2.

4.3 Poprawa klasyfikacji danych mniejszościowych

W ostatniej części badań meta-metod przeprowadzono testy w kierunku poprawy wykrywalności klasy mniejszościowej. W badaniach tych skupiono się wyłącznie na wpływie równoważenia liczebności klas w zbiorach danych na klasyfikację z wykorzystaniem meta-metod. Do badań wybrano klasyfikatory:

- bagging z drzewem decyzyjnym, z maksymalną głębokością 3 (Bag TREE),
- AdaBoost z drzewem decyzyjnym, z maksymalną głębokością 3 (AB Tree),
- stacking CV z drzewem decyzyjnym, kNN i naiwnym klasyfikatorem Bayesa oraz siecią neuronową MLP jako meta-klasyfikator (Stacking).

Baginng i AdaBoost zostały przetestowane z 50 klasyfikatorami. Ze względu na to, że jest to badanie w celu poprawy wykrywania klasy mniejszościowej, w wynikach zostały zaprezentowane tylko miara specyficzności oraz G-mean. Testy przeprowadzono zgodnie z opisem z rozdziału 3.4. Każdą klasyfikację powtórzono dziesięciokrotnie, a wyniki uśredniono. W sąsiednich kolumnach umieszczono meta-klasyfikator bez przetwarzania danych oraz z przetwarzaniem danych. Wyniki zostały porównane, większe wartości zostały pogrubione.

4.3.1 Oversampling metodą SMOTE

W pierwszym teście wygenerowano sztucznie dodatkowe dane klasy mniejszościowej metodą SMOTE. Przeprowadzone badanie znajduje się w pliku *imbalanced-tests/smote.py*. W przypadku metody bagging i stacking tylko w 3 zbiorach (zbiory z dużą liczbą przykładów bezpiecznych) lepszy okazał się klasyfikator bez metody SMOTE (tabela specyficzności 4.27). Użycie metody SMOTE pozwoliło zwiększyć wykrywalność klasy mniejszościowej o kilka procent, aż do ponad 100%. Każdemu wzrostowi specyficzności tych klasyfikatorów towarzyszył wzrost miary G-mean (tabela 4.28). Nieco gorzej z klasyfikacją poradził sobie model AdaBoost, w 14 na 22 zbiorach zanotowano wzrost. W zbiorach danych z dużą liczbą przykładów brzegowych, rzadkich i odstających zanotowano kilku procentowy spadek specyficzności dla klasyfikatora AdaBoost. Najlepszym meta-klasyfikatorem okazał się bagging z drzewem decyzyjnym i metodą SMOTE. Uzyskał on najlepszy wynik miary G-mean w 13 zbiorach. Prawie dla każdej bazy (20 z 22) zanotowano kilku procentowy spadek czułości klasy większościowej, tabeli z tymi wynikami nie zamieszczano w pracy.

Zbiór danych	Bag TREE	Bag TREE SMOTE	AB TREE	AB TREE SMOTE	Stacking	Stacking SMOTE
seeds	**0.86**	0.84	0.84	**0.86**	0.91	0.91
new_thyroid	**0.87**	0.81	**0.87**	0.86	**0.87**	0.84
vehicle	**0.92**	0.91	0.93	0.93	0.81	**0.85**
ionosphere	0.78	**0.82**	0.79	**0.83**	0.84	**0.87**
vertebal	0.75	**0.86**	0.72	**0.8**	0.75	**0.83**
yeastME3	0.77	**0.9**	0.63	**0.77**	0.77	**0.81**
ecoli	0.49	**0.81**	0.6	**0.67**	0.37	**0.83**
bupa	0.48	**0.58**	**0.56**	0.5	0.44	**0.57**
horse_colic	0.71	**0.76**	**0.76**	0.72	0.69	**0.71**
german	0.33	**0.61**	**0.52**	0.45	**0.41**	0.3
breast_cancer	0.29	**0.41**	**0.42**	0.35	0.33	**0.51**
cmc	0.25	**0.47**	**0.36**	0.34	**0.37**	0.29
hepatitis	0.47	**0.69**	0.59	**0.65**	0.44	**0.62**
haberman	0.28	**0.31**	**0.42**	0.39	0.14	**0.25**
transfusion	0.39	**0.62**	0.3	**0.33**	0.21	**0.45**
car	0.32	**0.67**	**0.65**	0.61	0.43	**0.59**
glass	0.0	**0.54**	0.24	**0.34**	0.0	**0.46**
abalone16_29	0.08	**0.77**	0.24	**0.3**	0.07	**0.59**
solar_flare	0.09	**0.72**	0.14	**0.21**	0.07	**0.53**
heart_cleveland	0.03	**0.41**	0.0	**0.08**	0.0	**0.24**
balance_scale	0.0	0.0	0.06	**0.09**	0.0	**0.18**
postoperative	0.04	**0.08**	0.17	**0.24**	0.0	**0.29**

Tablica 4.27: Specyficzność klasy mniejszościowej z użyciem metody SMOTE.

Zbiór danych	Bag TREE	Bag TREE SMOTE	AB TREE	AB TREE SMOTE	Stacking	Stacking SMOTE
seeds	**0.89**	0.88	0.88	**0.89**	0.92	0.92
new_thyroid	**0.92**	0.89	0.92	0.92	**0.92**	0.91
vehicle	**0.92**	0.9	**0.96**	0.95	0.88	**0.89**
ionosphere	0.86	**0.88**	0.87	**0.89**	0.89	**0.9**
vertebal	0.73	**0.78**	0.71	**0.75**	0.73	**0.76**
yeastME3	0.87	**0.92**	0.78	**0.86**	0.86	**0.88**
ecoli	0.67	**0.85**	0.75	**0.79**	0.59	**0.84**
bupa	0.66	**0.68**	**0.66**	0.61	0.59	**0.64**
horse_colic	0.81	**0.83**	**0.81**	0.79	0.81	**0.82**
german	0.56	**0.67**	**0.65**	0.61	**0.61**	0.51
breast_cancer	0.52	**0.59**	**0.57**	0.53	0.53	**0.61**
cmc	0.48	**0.59**	**0.55**	0.53	**0.58**	0.5
hepatitis	0.6	**0.68**	0.72	**0.73**	0.6	**0.66**
haberman	**0.51**	0.5	**0.55**	0.51	0.36	**0.46**
transfusion	0.59	**0.62**	**0.49**	0.47	0.44	**0.58**
car	0.48	**0.78**	**0.77**	0.73	0.63	**0.76**
glass	0.0	**0.57**	0.46	**0.52**	0.0	**0.6**
abalone16_29	0.28	**0.76**	0.48	**0.54**	0.27	**0.71**
solar_flare	0.3	**0.8**	0.37	**0.45**	0.26	**0.69**
heart_cleveland	0.17	**0.58**	0.0	**0.27**	0.0	**0.46**
balance_scale	0.0	0.0	0.24	**0.28**	0.0	**0.39**
postoperative	0.2	**0.26**	0.36	**0.41**	0.0	**0.44**

Tablica 4.28: Miara G-mean z użyciem metody SMOTE.

4.3.2 Oversampling metodą ADASYN

Metoda ADASYN to zmodyfikowana wersja algorytmu SMOTE, generująca nowe próbki głównie w okolicach przykładów trudnych w klasyfikacji. Test z wykorzystaniem tej metody znajduje się w pliku *imbalancedtests/adasyn.py*. Skuteczność klasyfikacji klasy mniejszościowej (tabela specyficzności 4.29) z wykorzystaniem tej metody wzrosła o kilka procent dla wszystkich klasyfikatorów i prawie wszystkich baz danych. Bagging z metodą ADASYN dla baz *solar_flare* oraz *abalone16_29* podniósł wykrywalność klasy mniejszościowej z poziomu poniżej 10% do powyżej 80%. Każdy meta-klasyfikator uzyskał gorszą specyficzność z metodą ADASYN dla bazy *new thyroid* oraz *german* (podobnie jak w metodzie SMOTE). Bazy te zawierają dużo przykładów brzegowych, rzadkich i odstających.

Zbiór danych	Bag TREE	Bag TREE ADASYN	AB TREE	AB TREE ADASYN	Stacking	Stacking ADASYN
seeds	0.87	**0.93**	0.84	**0.91**	0.86	**0.96**
new_thyroid	**0.87**	0.86	**0.87**	0.83	**0.87**	0.84
vehicle	**0.94**	0.92	0.94	**0.96**	0.82	**0.85**
ionosphere	0.77	**0.83**	0.79	**0.82**	0.79	**0.85**
vertebal	0.73	**0.91**	0.77	**0.79**	0.77	**0.89**
yeastME3	0.77	**0.96**	0.63	**0.82**	0.71	**0.86**
ecoli	0.49	**0.94**	0.6	**0.76**	0.37	**0.86**
bupa	0.43	**0.62**	0.55	**0.58**	0.39	**0.54**
horse_colic	0.74	**0.75**	0.74	**0.79**	0.68	**0.74**
german	**0.33**	0.15	**0.51**	0.47	**0.37**	0.2
breast_cancer	0.33	**0.41**	0.41	**0.46**	0.25	**0.54**
cmc	0.25	**0.53**	0.36	**0.38**	0.14	**0.4**
hepatitis	0.53	**0.58**	0.59	**0.64**	0.34	**0.64**
haberman	0.27	**0.49**	0.42	**0.49**	0.01	**0.59**
transfusion	0.36	**0.59**	0.3	**0.4**	0.18	**0.59**
car	0.32	**0.78**	0.6	0.6	0.43	**0.58**
glass	0.0	**0.44**	0.18	**0.28**	0.12	**0.39**
abalone16_29	0.08	**0.81**	0.23	**0.35**	0.06	**0.49**
solar_flare	0.09	**0.84**	0.09	**0.23**	0.0	**0.59**
heart_cleveland	0.03	**0.45**	0.0	**0.11**	0.0	**0.4**
balance_scale	0.0	**0.18**	0.06	**0.33**	0.0	**0.41**
postoperative	0.04	**0.06**	0.17	**0.21**	0.12	**0.32**

Tablica 4.29: Specyficzność klasy mniejszościowej z metodą ADASYN.

Zanotowano także wzrost miary G-mean (tabela 4.30) w 19 bazach dla każdego meta-klasyfikatora. W efekcie wzrostu specyficzności klasy mniejszościowej, pogorszyła się dokładności klasyfikacji oraz spadła czułość klasy dominującej. Spadki te wyniosły najczęściej od kilku procent do 20%.

Zbiór danych	Bag TREE	Bag TREE ADASYN	AB TREE	AB TREE ADASYN	Stacking	Stacking ADASYN
seeds	**0.9**	0.88	0.89	**0.9**	**0.89**	0.88
new_thyroid	0.92	0.92	**0.92**	0.91	**0.92**	0.9
vehicle	**0.93**	0.91	0.96	**0.97**	0.88	**0.89**
ionosphere	0.86	**0.87**	0.87	**0.89**	0.86	**0.88**
vertebal	0.71	**0.79**	**0.73**	0.71	0.74	**0.77**
yeastME3	0.86	**0.93**	0.78	**0.88**	0.83	**0.88**
ecoli	0.68	**0.87**	0.75	**0.81**	0.59	**0.86**
bupa	0.62	**0.67**	0.64	**0.66**	0.59	0.59
horse_colic	0.83	0.83	0.81	**0.82**	0.8	**0.82**
german	**0.55**	0.37	**0.64**	0.62	**0.59**	0.41
breast_cancer	0.54	**0.58**	0.56	**0.59**	0.48	**0.63**
cmc	0.48	**0.64**	0.55	**0.56**	0.37	**0.59**
hepatitis	0.62	**0.63**	0.72	**0.73**	0.51	**0.69**
haberman	0.5	**0.62**	0.55	**0.59**	0.11	**0.61**
transfusion	0.56	**0.6**	0.49	**0.52**	0.42	**0.59**
car	0.48	**0.84**	0.74	**0.75**	0.63	**0.74**
glass	0.0	**0.52**	0.4	**0.47**	0.33	**0.53**
abalone16_29	0.28	**0.77**	0.48	**0.57**	0.24	**0.66**
solar_flare	0.3	**0.83**	0.3	**0.46**	0.0	**0.71**
heart_cleveland	0.17	**0.62**	0.0	**0.31**	0.0	**0.58**
balance_scale	0.0	**0.38**	0.24	**0.47**	0.0	**0.51**
postoperative	0.2	**0.23**	0.37	**0.4**	0.34	**0.4**

Tablica 4.30: Miara G-mean z użyciem metody ADASYN.

4.3.3 Undersampling NCR

Test meta-klasyfikatorów z metodą NCR został przeprowadzony tak samo jak poprzednie i znajduje się w pliku *imbalancedtests/NCR.py*. Wszystkie meta-klasyfikatory poprawiły wynik specyficzności (tabela specyficzności 4.31) dla 20 baz.

Zbiór danych	Bag TREE	Bag TREE NCR	AB TREE	AB TREE NCR	Stacking	Stacking NCR
seeds	0.84	**0.93**	0.84	**0.92**	0.9	**0.96**
new_thyroid	0.87	**0.93**	0.87	**0.88**	0.87	**0.95**
vehicle	0.9	**0.98**	0.93	**0.97**	0.85	**0.89**
ionosphere	0.79	**0.81**	0.8	**0.83**	0.84	**0.87**
vertebal	0.76	**0.93**	0.72	**0.89**	0.71	**0.92**
yeastME3	0.75	**0.85**	0.63	**0.82**	0.69	**0.79**
ecoli	0.51	**0.67**	0.6	**0.69**	0.54	**0.69**
bupa	0.43	**0.82**	0.56	**0.82**	0.41	**0.79**
horse_colic	0.73	**0.85**	0.71	**0.82**	0.7	**0.8**
german	0.34	**0.77**	0.52	**0.67**	0.42	**0.7**
breast_cancer	0.33	**0.64**	0.42	**0.63**	0.28	**0.63**
cmc	0.24	**0.43**	0.36	**0.58**	0.17	**0.51**
hepatitis	0.53	**0.66**	0.59	**0.7**	**0.44**	0.4
haberman	0.3	**0.5**	0.42	**0.68**	0.14	**0.45**
transfusion	0.36	**0.58**	0.3	**0.56**	0.27	**0.52**
car	0.32	0.32	**0.65**	0.63	0.43	**0.45**
glass	0.0	**0.01**	**0.29**	0.27	0.0	**0.06**
abalone16_29	0.08	**0.13**	0.24	**0.34**	0.06	**0.23**
solar_flare	0.09	**0.16**	0.12	**0.41**	0.0	**0.26**
heart_cleveland	0.0	**0.15**	0.03	**0.08**	0.0	**0.07**
balance_scale	0.0	0.0	0.06	**0.07**	0.0	0.0
postoperative	0.04	**0.23**	0.21	**0.5**	0.04	**0.23**

Tablica 4.31: Specyficzność klasy mniejszościowej z metodą NCR.

Podobnie jak w poprzednich metodach, wzrost wyniósł średnio kilka procent. Zanotowano mniejszy spadek dokładności i czułości niż w poprzednich metodach. Usunięcie przykładów z klasy mniejszościowej nie zwiększyło specyficzności w trudnych bazach *glass* oraz *balanace scale* i klasyfikatory nadal nie wykrywały przykładów klasy mniejszościowej w tych bazach. Miara G-mean (tabela 4.32) wzrosła dla wszystkich baz oprócz *bupa* oraz *horse_ colic*. Jest to związane z tym, że dla tych dwóch baz czułość klasy większościowej spadła odpowiednio o 20% oraz o 50%. Obie bazy zawierają dużo przykładów granicznych, dla których NCR mógł usunąć z sąsiedztwa obserwacje klasy dominującej.

Zbiór danych	Bag TREE	Bag TREE NCR	AB TREE	AB TREE NCR	Stacking	Stacking NCR
seeds	0.88	**0.9**	0.88	**0.89**	0.9	**0.92**
new_thyroid	0.92	**0.95**	0.92	**0.93**	0.92	**0.96**
vehicle	0.92	**0.93**	0.96	0.96	0.89	0.89
ionosphere	0.86	**0.88**	0.89	0.89	0.86	**0.9**
vertebal	0.73	**0.8**	0.71	**0.77**	0.71	**0.79**
yeastME3	0.85	**0.9**	0.78	**0.89**	0.82	**0.87**
ecoli	0.69	**0.73**	0.75	**0.76**	0.71	**0.79**
bupa	**0.62**	0.58	**0.67**	0.63	**0.58**	0.57
horse_colic	**0.82**	0.8	**0.79**	0.75	**0.82**	0.76
german	0.56	**0.71**	0.65	**0.66**	0.61	**0.69**
breast_cancer	0.54	**0.62**	0.57	**0.59**	0.5	**0.65**
cmc	0.47	**0.6**	0.55	**0.62**	0.41	**0.63**
hepatitis	0.65	**0.71**	0.72	**0.74**	**0.61**	0.58
haberman	0.52	**0.63**	0.55	**0.61**	0.36	**0.59**
transfusion	0.57	**0.63**	0.49	**0.58**	0.49	**0.58**
car	0.48	0.48	0.77	**0.79**	0.63	**0.64**
glass	0.0	**0.02**	**0.51**	0.47	0.0	**0.19**
abalone16_29	0.28	**0.36**	0.48	**0.57**	0.25	**0.47**
solar_flare	0.3	**0.39**	0.34	**0.61**	0.0	**0.5**
heart_cleveland	0.0	**0.38**	0.16	**0.27**	0.0	**0.23**
balance_scale	0.0	0.0	0.24	**0.26**	0.0	0.0
postoperative	0.2	**0.39**	0.39	**0.41**	0.2	**0.38**

Tablica 4.32: Miara G-mean z metodą NCR.

4.3.4 Oversampling SMOTE i undersampling metodą ENN

W kolejnym teście znajdującym się w pliku *imbalancedtests/smoteeen.py*, do równoważenia klas w zbiorach danych wybrano metodę SMOTEENN. W pierwszej kolejności algorytm generuje sztuczne obserwacje metodą SMOTE, a następnie usuwa przykłady z klasy większościowej metodą ENN oraz sztuczne przykłady z klasy mniejszościowej, jeżeli ingerują w przestrzeń klasy dominującej. Otrzymane wyniki specyficzności są zdecydowanie gorsze niż dla poprzednich metod. Tylko Stacking ze SMOTEENN uzyskał w 14 bazach lepsze wyniki, pozostałe klasyfikatory uzyskały lepsze wyniki dla połowy zbiorów lub mniej. Specyficzność (tabela 4.33) zmalała w bazach zawierających dużo przykładów bezpiecznych (*seeds*, *new thyroid*, *vehicle*, *ionosphere*), z dużą liczbą przykładów granicznych (*bupa*, *horse_ colic*, *german*) oraz z dużą liczbą przykładów mieszanych niebezpiecznych (*haberman*, *transfusion*). Porównując wyniki z samą metodą SMOTE, widać pogorszenie skuteczności klasyfi-

kacji dla połączonych metod SMOTE i ENN. W zbiorach, w których spadła specyficzność klasy mniejszościowej, zwiększyła się wykrywalność klasy większościowej. Miarę G-mean przedstawiono w tabeli 4.34. Meta-klasyfikator bagging aż w połowie zbiorów danych osiągnął gorszy wynik w klasyfikacji obu klas.

Zbiór danych	Bag TREE	Bag TREE SMOTEENN	AB TREE	AB TREE SMOTEENN	Stacking	Stacking SMOTEENN
seeds	**0.89**	0.82	**0.89**	0.83	**0.93**	0.88
new_thyroid	**0.87**	0.76	0.87	0.87	**0.93**	0.84
vehicle	**0.92**	0.83	**0.93**	0.81	**0.84**	0.72
ionosphere	**0.8**	0.74	**0.79**	0.72	**0.81**	0.74
vertebal	**0.75**	0.74	**0.74**	0.73	0.75	**0.78**
yeastME3	0.77	**0.9**	0.63	**0.77**	0.72	**0.82**
ecoli	0.49	**0.82**	0.57	**0.67**	0.57	**0.79**
bupa	**0.43**	0.32	**0.53**	0.27	**0.36**	0.33
horse_colic	**0.74**	0.66	**0.71**	0.55	**0.71**	0.58
german	**0.38**	0.01	**0.51**	0.07	**0.45**	0.04
breast_cancer	**0.31**	0.23	**0.42**	0.22	0.24	**0.27**
cmc	0.28	0.28	**0.36**	0.12	0.15	**0.18**
hepatitis	0.56	**0.61**	**0.59**	0.56	0.44	**0.52**
haberman	**0.31**	0.01	**0.42**	0.19	0.05	**0.09**
transfusion	**0.42**	0.0	**0.3**	0.18	**0.22**	0.18
car	0.32	**0.69**	**0.65**	0.61	0.43	**0.59**
glass	0.0	**0.53**	0.06	**0.32**	0.12	**0.45**
abalone16_29	0.08	**0.75**	0.24	**0.33**	0.09	**0.56**
solar_flare	0.09	**0.72**	0.12	**0.13**	0.0	**0.43**
heart_cleveland	0.03	**0.37**	0.03	**0.04**	0.03	**0.19**
balance_scale	0.0	0.0	**0.06**	0.02	0.0	**0.18**
postoperative	0.04	**0.06**	**0.21**	0.07	0.0	**0.08**

Tablica 4.33: Specyficzność klasy mniejszościowej z metodą SMOTEENN.

Zbiór danych	Bag TREE	Bag TREE SMOTEENN	AB TREE	AB TREE SMOTEENN	Stacking	Stacking SMOTEENN
seeds	**0.91**	0.88	**0.91**	0.89	**0.92**	0.9
new_thyroid	**0.92**	0.86	0.92	0.92	**0.95**	0.91
vehicle	**0.92**	0.88	**0.96**	0.89	**0.89**	0.83
ionosphere	**0.87**	0.85	**0.87**	0.84	**0.87**	0.85
vertebal	0.72	**0.73**	0.72	**0.73**	0.73	**0.75**
yeastME3	0.86	**0.92**	0.78	**0.86**	0.84	**0.89**
ecoli	0.67	**0.86**	0.73	**0.78**	0.73	**0.83**
bupa	**0.63**	0.55	**0.64**	0.51	**0.56**	0.54
horse_colic	**0.82**	0.79	**0.79**	0.72	**0.81**	0.73
german	**0.59**	0.1	**0.64**	0.27	**0.63**	0.19
breast_cancer	**0.52**	0.46	**0.57**	0.45	0.47	**0.49**
cmc	**0.51**	0.49	**0.55**	0.34	0.38	**0.4**
hepatitis	0.67	**0.69**	**0.73**	0.7	0.61	**0.64**
haberman	**0.53**	0.1	**0.55**	0.42	0.22	**0.29**
transfusion	**0.61**	0.0	**0.5**	0.4	**0.44**	0.41
car	0.48	**0.79**	**0.77**	0.73	0.63	**0.75**
glass	0.0	**0.57**	0.23	**0.51**	0.33	**0.59**
abalone16_29	0.28	**0.76**	0.48	**0.56**	0.3	**0.7**
solar_flare	0.3	**0.8**	0.34	**0.36**	0.0	**0.63**
heart_cleveland	0.17	**0.56**	0.16	**0.17**	0.17	**0.41**
balance_scale	0.0	0.0	**0.24**	0.11	0.0	**0.4**
postoperative	0.2	**0.24**	**0.39**	0.25	0.0	**0.27**

Tablica 4.34: Miara G-mean z metodą SMOTEENN.

4.3.5 Oversampling SMOTE i undersampling metodą Tomek links

W następnym teście zbadano wpływ połączonej metody SMOTE z usuwaniem przykładów metodą Tomek links. Test zamieszczono w pliku *imbalancedtests/SMOTETomek.py*. Z wykorzystaniem metody SMOTETomek (w tabeli wyniki z tą metodą oznaczone są jako SMOTETomek) uzyskano zdecydowanie lepsze wyniki niż w metodzie SMOTEENN (tabela specyficzności 4.35). Metoda ta poprawiła wyniki w baggingu aż w 19 zbiorach, w stackingu w 17 zbiorach, a w algorytmie AdaBoost w 14 zbiorach. Kolejny raz nie udało się poprawić wyników w zbiorach z dużą liczbą bezpiecznych przykładów (*seeds*, *new thyroid*, *vehicle*) oraz w zbiorach (*german* oraz *haberman*).

Zbiór danych	Bag TREE	Bag TREE SMOTET	AB TREE	AB TREE SMOTET	Stacking	Stacking SMOTET
seeds	0.84	**0.85**	**0.89**	0.87	**0.91**	0.9
new_thyroid	**0.87**	0.81	**0.87**	0.86	**0.93**	0.84
vehicle	0.91	0.91	**0.95**	0.89	**0.88**	0.83
ionosphere	0.78	**0.81**	0.81	**0.82**	**0.87**	0.86
vertebal	0.77	**0.86**	0.75	**0.8**	0.68	**0.81**
yeastME3	0.74	**0.9**	0.63	**0.76**	0.7	**0.82**
ecoli	0.51	**0.81**	0.51	**0.68**	0.43	**0.83**
bupa	0.43	**0.47**	**0.55**	0.48	0.5	0.5
horse_colic	0.74	**0.76**	**0.75**	0.73	0.67	**0.72**
german	**0.34**	0.33	**0.51**	0.43	**0.29**	0.1
breast_cancer	0.31	**0.38**	0.41	0.41	0.27	**0.4**
cmc	0.24	**0.47**	**0.36**	0.3	0.16	**0.29**
hepatitis	0.5	**0.68**	0.62	**0.63**	0.34	**0.62**
haberman	**0.31**	0.25	**0.42**	0.25	0.19	**0.21**
transfusion	0.37	**0.45**	0.3	**0.34**	0.08	**0.4**
car	0.32	**0.69**	**0.65**	0.61	0.43	**0.59**
glass	0.0	**0.53**	0.12	**0.31**	0.06	**0.45**
abalone16_29	0.08	**0.78**	0.24	**0.3**	0.1	**0.59**
solar_flare	0.12	**0.72**	0.12	**0.21**	0.05	**0.52**
heart_cleveland	0.06	**0.39**	0.0	**0.03**	0.0	**0.26**
balance_scale	0.0	0.0	0.06	**0.1**	0.0	**0.18**
postoperative	0.04	**0.09**	0.17	**0.25**	0.0	**0.33**

Tablica 4.35: Specyficzność klasy mniejszościowej dla metody SMOTE z Tomek links.

Wartość miary G-mean (tabela 4.36) oprócz wymienionych wcześniej zbiorów wzrosła, mimo że czułość klasy większościowej spadła dla wszystkich zbiorów, w których zanotowano wzrost wykrywalności klasy zdominowanej. Undersampling Tomek links usuwa szum oraz „czyści" granicę pomiędzy klasami, zwiększając obszar klasy mniejszościowej. W efekcie rośnie rozpoznawalność klasy zdominowanej.

Zbiór danych	Bag TREE	Bag TREE SMOTET	AB TREE	AB TREE SMOTET	Stacking	Stacking SMOTET
seeds	0.88	**0.89**	0.9	0.9	**0.92**	0.91
new_thyroid	**0.92**	0.9	0.92	0.92	**0.95**	0.91
vehicle	**0.91**	0.9	**0.97**	0.93	**0.9**	0.88
ionosphere	0.86	**0.88**	0.88	**0.89**	0.89	**0.9**
vertebal	0.75	**0.78**	0.72	**0.75**	0.71	**0.76**
yeastME3	0.85	**0.92**	0.78	**0.85**	0.82	**0.88**
ecoli	0.69	**0.85**	0.69	**0.79**	0.64	**0.85**
bupa	0.62	**0.63**	**0.65**	0.62	0.63	0.63
horse_colic	0.82	**0.83**	0.81	0.81	0.8	**0.82**
german	**0.56**	0.54	**0.64**	0.61	**0.52**	0.29
breast_cancer	0.52	**0.57**	0.56	**0.58**	0.5	**0.57**
cmc	0.47	**0.59**	**0.55**	0.5	0.39	**0.5**
hepatitis	0.64	**0.68**	0.73	0.73	0.52	**0.67**
haberman	**0.53**	0.47	**0.55**	0.44	0.42	**0.43**
transfusion	0.57	0.57	0.49	0.49	0.28	**0.55**
car	0.48	**0.79**	**0.77**	0.73	0.63	**0.76**
glass	0.0	**0.58**	0.33	**0.5**	0.23	**0.58**
abalone16_29	0.28	**0.76**	0.48	**0.54**	0.31	**0.71**
solar_flare	0.34	**0.79**	0.34	**0.45**	0.21	**0.69**
heart_cleveland	0.24	**0.57**	0.0	**0.12**	0.0	**0.47**
balance_scale	0.0	0.0	0.24	**0.3**	0.0	**0.39**
postoperative	0.2	**0.27**	0.35	**0.44**	0.0	**0.47**

Tablica 4.36: Miara G-mean SMOTE z Tomek links.

4.3.6 Porównanie metod

W ostatnim teście meta-metod z wstępnym przetwarzaniem zbiorów, dokonano zbiorczego porównania wszystkich metod. Zrównoważone klasy w zbiorach były klasyfikowane z wykorzystaniem meta-klasyfikatora bagging z drzewem decyzyjnym (z ustawioną maksymalną głębokością drzewa 3). Test ten znajduje się w pliku *imbalancedtests/compare_methods.py*. W tabeli 4.37 została przedstawiona uzyskana dokładność klasyfikatorów z różnymi metodami. Zdecydowanie najlepszą dokładność w większości zbiorów osiągnął klasyfikator bagging uczony na oryginalnych danych (pierwsza kolumna). Algorytm SMOTEENN, w którym usuwane są przykłady z klasy większościowej mające w większości sąsiadów z klasy mniejszościowej, poprawił o kilka procent wykrywalność klasy większościowej (tabela 4.38). W pozostałych zbiorach, klasyfikatorem z największą czułością był bagging. Algorytmy równoważenia zbiorów zostały stworzone w celu poprawy wykrywalności klasy mniejszościowej i ten cel spełniają. Specyficzność klasy zdominowanej (tabela 4.39) oraz miara G-mean (tabela 4.40) zwiększyła się prawie we wszystkich zbiorach (w jednym przypadku została na tym samym poziomie). Najlepszym algorytmem okazał się ADASYN oraz NCR. Algorytm NCR jest skuteczny głównie dla zbiorów z duża liczbą przykładów bezpiecznych i granicznych. W zbiorach z dużą liczbą przykładów rzadkich i odstających, wzrost jakości klasyfikacji był minimalny. Należy zwrócić uwagę na zbiór *glass*, w którym dokładność klasyfikacji po zrównoważeniu zbiorów spadła o ponad 30%. Jednak wykrywalność klasy zdominowanej wzrosła z poziomu zerowego do ponad 60%.

Zbiór danych	Bag	SMOTE	ADASYN	NCR	SMOTEENN	SMOTET
seeds	**0.9**	0.89	0.86	0.89	**0.9**	**0.9**
new_thyroid	**0.97**	0.96	0.96	**0.97**	0.96	0.96
vehicle	**0.92**	0.89	0.9	**0.92**	0.91	0.89
ionosphere	0.89	**0.9**	0.88	**0.9**	0.89	**0.9**
vertebal	0.72	**0.76**	**0.76**	**0.76**	0.73	**0.76**
yeastME3	**0.95**	0.94	0.9	**0.95**	0.94	0.94
ecoli	0.87	**0.89**	0.8	0.79	**0.89**	**0.89**
bupa	**0.71**	**0.71**	0.68	0.58	0.68	0.7
horse_colic	**0.86**	0.85	0.85	0.79	0.84	0.85
german	**0.75**	0.7	0.71	0.69	0.7	0.72
breast_cancer	**0.72**	0.71	0.71	0.61	0.71	**0.72**
cmc	**0.77**	0.69	0.71	0.76	0.72	0.69
hepatitis	0.71	0.68	0.67	**0.74**	**0.74**	0.69
haberman	**0.74**	0.7	0.71	0.72	0.72	0.72
transfusion	**0.78**	0.62	0.61	0.65	0.76	0.66
car	0.69	**0.9**	**0.9**	0.69	**0.9**	**0.9**
glass	**0.9**	0.61	0.58	0.89	0.61	0.61
abalone16_29	**0.94**	0.75	0.73	**0.94**	0.77	0.76
solar_flare	**0.95**	0.87	0.82	0.94	0.87	0.87
heart_cleveland	**0.86**	0.78	0.81	0.85	0.79	0.78
balance_scale	**0.92**	**0.92**	0.75	**0.92**	**0.92**	**0.92**
postoperative	0.71	0.62	0.66	0.53	**0.72**	0.64

Tablica 4.37: Dokładność klasyfikacji - porównanie metod równoważenia liczebności klas

Zbiór danych	Bag	SMOTE	ADASYN	NCR	SMOTEENN	SMOTET
seeds	**0.93**	0.92	0.82	0.87	**0.93**	**0.93**
new_thyroid	0.98	**0.99**	0.98	0.98	**0.99**	**0.99**
vehicle	0.92	0.88	0.9	0.9	**0.93**	0.88
ionosphere	0.95	0.94	0.91	0.94	**0.98**	0.94
vertebal	0.7	0.71	0.69	0.68	**0.72**	**0.72**
yeastME3	**0.97**	0.94	0.89	0.96	0.94	0.94
ecoli	**0.91**	0.9	0.78	0.8	0.9	0.9
bupa	0.87	0.81	0.73	0.41	**0.94**	0.85
horse_colic	0.92	0.9	0.91	0.76	**0.95**	0.9
german	0.92	0.73	0.94	0.65	**1.0**	0.88
breast_cancer	0.9	0.84	0.84	0.59	**0.92**	0.86
cmc	**0.93**	0.76	0.76	0.85	0.84	0.76
hepatitis	0.76	0.67	0.69	0.77	**0.78**	0.69
haberman	0.9	0.84	0.79	0.81	**0.97**	0.89
transfusion	0.9	0.62	0.61	0.67	**1.0**	0.73
car	0.71	**0.91**	0.9	0.71	**0.91**	0.9
glass	**0.97**	0.62	0.59	0.96	0.61	0.62
abalone16_29	**1.0**	0.75	0.72	0.99	0.77	0.76
solar_flare	**0.99**	0.88	0.82	0.98	0.88	0.88
heart_cleveland	**0.97**	0.83	0.85	0.94	0.84	0.84
balance_scale	**1.0**	**1.0**	0.8	**1.0**	**1.0**	**1.0**
postoperative	0.94	0.82	0.88	0.64	**0.97**	0.83

Tablica 4.38: Czułość klasy większościowej - porównanie metod równoważenia liczebności klas

Zbiór danych	Bag	SMOTE	ADASYN	NCR	SMOTEENN	SMOTET
seeds	0.84	0.83	**0.93**	**0.93**	0.84	0.84
new_thyroid	0.87	0.81	0.86	**0.92**	0.76	0.79
vehicle	0.92	0.91	0.9	**0.98**	0.84	0.9
ionosphere	0.79	0.82	**0.83**	0.82	0.74	0.82
vertebal	0.75	0.86	0.9	**0.93**	0.74	0.86
yeastME3	0.77	0.9	**0.96**	0.84	0.9	0.91
ecoli	0.49	0.81	**0.94**	0.67	0.81	0.81
bupa	0.48	0.58	0.62	**0.84**	0.31	0.5
horse_colic	0.75	0.77	0.75	**0.85**	0.65	0.76
german	0.36	0.63	0.2	**0.77**	0.02	0.33
breast_cancer	0.29	0.41	0.41	**0.65**	0.23	0.38
cmc	0.21	0.46	**0.54**	0.43	0.3	0.46
hepatitis	0.53	**0.69**	0.57	0.66	0.61	0.68
haberman	0.28	0.32	**0.49**	**0.49**	0.01	0.25
transfusion	0.37	**0.63**	0.59	0.58	0.0	0.46
car	0.32	0.67	**0.88**	0.32	0.69	0.72
glass	0.0	**0.53**	0.46	0.01	**0.53**	0.52
abalone16_29	0.08	0.77	**0.81**	0.14	0.75	0.78
solar_flare	0.09	0.72	**0.84**	0.15	0.7	0.72
heart_cleveland	0.03	0.4	**0.44**	0.15	0.38	0.39
balance_scale	0.0	0.0	**0.18**	0.0	0.0	0.0
postoperative	0.08	0.09	0.06	**0.22**	0.05	0.09

Tablica 4.39: Specyficzność klasy mniejszościowej - porównanie metod równoważenia liczebności klas

Zbiór danych	Bag	SMOTE	ADASYN	NCR	SMOTEENN	SMOTET
seeds	0.88	0.88	0.87	**0.9**	0.88	0.88
new_thyroid	0.92	0.89	0.92	**0.95**	0.86	0.89
vehicle	0.92	0.89	0.9	**0.94**	0.88	0.89
ionosphere	0.87	**0.88**	0.87	**0.88**	0.85	**0.88**
vertebal	0.73	0.78	**0.79**	**0.79**	0.73	0.78
yeastME3	0.86	0.92	**0.93**	0.9	0.92	0.92
ecoli	0.67	0.85	**0.86**	0.73	0.85	0.85
bupa	0.64	**0.68**	0.67	0.58	0.54	0.65
horse_colic	**0.83**	**0.83**	**0.83**	0.8	0.79	**0.83**
german	0.57	0.68	0.43	**0.71**	0.09	0.54
breast_cancer	0.51	0.59	0.59	**0.62**	0.46	0.57
cmc	0.44	0.59	**0.64**	0.6	0.5	0.59
hepatitis	0.63	0.68	0.63	**0.71**	0.69	0.69
haberman	0.51	0.52	0.62	**0.63**	0.08	0.47
transfusion	0.58	**0.62**	0.6	**0.62**	0.0	0.57
car	0.48	0.78	**0.88**	0.48	0.79	0.8
glass	0.0	**0.57**	0.52	0.02	**0.57**	**0.57**
abalone16_29	0.29	0.76	**0.77**	0.37	0.76	**0.77**
solar_flare	0.3	0.8	**0.83**	0.38	0.79	0.8
heart_cleveland	0.17	0.57	**0.61**	0.37	0.56	0.57
balance_scale	0.0	0.0	**0.38**	0.0	0.0	0.0
postoperative	0.28	0.27	0.22	**0.37**	0.22	0.28

Tablica 4.40: Miara G-mean - porównanie metod równoważenia liczebności klas

4.4 Wnioski z badań

Stosując meta-metody można podnieść o kilka procent skuteczność klasyfikacji danych. Najlepszym meta-klasyfikatorem okazał się bagging z naiwnym klasyfikatorem Bayesa poprawiając klasyfikację w większości zbiorów danych. Najbardziej stabilne wyniki dla wszystkich baz otrzymano z meta-klasyfikatora stacking. AdaBoost uzyskał minimalnie gorsze wyniki.

Aby uzyskać dobre wyniki w baggingu, należy znaleźć optymalne ustawienia liczby atrybutów oraz wielkości podzbiorów. Las losowy uzyskał bardzo dobre wyniki klasyfikacji. Jest to specjalny przypadek baggingu o specyficznych ustawieniach. Bagging dla wielkości podzbiorów mniejszych niż liczebność oryginalnego zbioru oraz dla mniejszej liczby atrybutów uzyskuje lepsze wyniki. Powyżej 50 klasyfikatorów przyrost dokładności klasyfikacji był minimalny, spadła natomiast specyficzność klasy mniejszościowej.

Meta-klasyfikator AdaBoost najlepsze wyniki uzyskiwał dla słabego klasyfikatora np. drzewa decyzyjnego z głębokością 3. Natomiast używając AdaBoost z naiwnym klasyfikatorem Bayesa wyniki klasyfikacji pogorszyły się. Im podstawowy klasyfikator był mocniejszy tym przyrost skuteczności był mniejszy, a w przypadku dobrego klasyfikatora występował wzrost błędu klasyfikacji.

Bagging oparty o algorytm kNN uzyskał wysoką dokładność klasyfikacji oraz lepszą skuteczność klasyfikacji klasy większościowej. Pomimo poprawienia dokładności klasyfikacji, wykrywalność klasy mniejszościowej była na niskim poziomi.

Bagging i boosting zbudowane z naiwnego klasyfikatora Bayesa wykazywały się najlepszą skutecznością klasyfikacji klasy mniejszościowej. Efektem zwiększenia specyficzności klasy zdominowanej, była duża liczba fałszywych alarmów i niska precyzja klasyfikacji tej klasy (duża liczba błędnie zakwalifikowanych przykładów do klasy mniejszościowej).

W zbiorach zawierających dużą liczbę przykładów bezpiecznych, przyrost skuteczności klasyfikacji obu klas był minimalny.

Zastosowanie metod równoważenia liczebności klasy z meta-metodami zwiększa skuteczność klasyfikacji klasy zdominowanej, jednocześnie pogarszając czułość klasy większościowej. Najlepsze wyniki uzyskano z wykorzystaniem algorytmów ADASYN oraz NCR.

Rozdział 5

Propozycja klasyfikatorów

5.1 Klasyfikator ekspercki

Klasyfikator ekspercki powstał na bazie doświadczeń z podstawowymi klasyfikatorami. W zależności od charakterystyki danych, osiągana skuteczność przez klasyfikatory może się różnić. Klasyfikator skutecznie rozpoznający klasy jednego zbioru może miernie klasyfikować inny zbiór, podczas gdy użycie innego klasyfikatora na tym samym zbiorze danych może znacząco poprawić osiągane wyniki. Również użycie różnych algorytmów klasyfikacji, połączenie ich w komitet może zmniejszyć błąd klasyfikacji. Klasyfikator ekspercki został stworzony w celu zmniejszenia błędu klasyfikacji, niezależnie od typu danych oraz nieznanych danych.

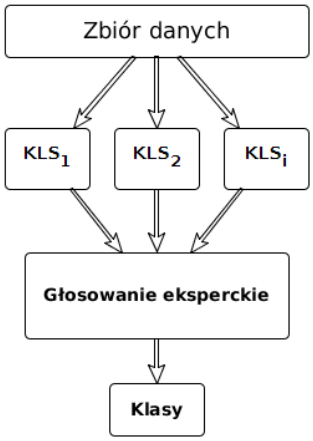
Klasyfikator ekspercki to połączenie kilku klasyfikatorów (przynajmniej trzech), najlepiej z różnymi algorytmami lub ustawieniami. Każdy klasyfikator trenowany jest na zbiorze uczącym, a następnie oceniana jest jakość klasyfikacji każdego z osobna. Stworzono dwie wersje klasyfikatora, w pierwszej klasyfikator uczony i testowany jest na tym samym zbiorze danych, w drugiej oceniany jest z wykorzystaniem sprawdzianu krzyżowego (domyślnie k=3). W kolejnym etapie, dla każdej klasy wyłaniany jest klasyfikator ekspert. Ekspert klasowy wybierany jest na podstawie najwyższego współczynnika dla danej klasy. Tworząc klasyfikator można wybrać na podstawie którego współczynnika precyzji, F1 czy G-mean będą wyłaniani eksperci. Domyślnie jest to współczynnik precyzji. Przy wyborze miary G-mean ekspert dla obu będzie taki sam. Jeżeli ocena odbywała się ze sprawdzianem krzyżowym, to modele klasyfikatora tworzone są od nowa, na całym zbiorze treningowym.

W procesie klasyfikacji właściwej, nowe próbki klasyfikowane są najpierw przez klasyfikatory składowe. Końcowa klasa wyznaczana jest według algorytmu:

1. Jeżeli występuje zgodność co do klasy pomiędzy klasyfikatorami, to wybierana jest ta klasa.
2. Jeżeli tylko jeden ekspert wskaże swoją klasę, to ostateczną klasą jest ta wskazana przez eksperta.
3. Jeżeli dwóch ekspertów wskaże swoje klasy, to wybierana jest klasa z większym

prawdopodobieństwem wskazanym przez klasyfikator. W przypadku takich samych prawdopodobieństw, wybierana jest klasa wskazana przez klasyfikator z większym współczynnikiem G-mean.

4. Jeżeli żaden ekspert nie wskaże swojej klasy, to klasa wybierana jest poprzez głosowanie większościowe.



Rysunek 5.1: Schemat klasyfikatora eksperckiego. $KLS_1..KLS_i$ to klasyfikatory składowe.

Podstawowa wersja klasyfikatora znajduje się w pliku $classifiers/clf\_expert.py$, a wersja ze sprawdzianem krzyżowym w pliku $classifiers/clf\_expertCV.py$. Obsługa klasyfikatora odbywa się w takim sam sposób jak klasyfikatorów z biblioteki $scikit-learn$.

5.1.1 Test klasyfikatora eksperckiego

Test klasyfikatora eksperckiego przeprowadzono tak samo jak poprzednie. Został on powtórzony 10 razy, a wyniki zostały uśrednione. Test znajduje się w pliku $test\_my\_clfs/test\_clf\_expert.py$. Klasyfikator ekspercki został zbudowany z naiwnego klasyfikatora Bayesa, klasyfikatora kNN oraz drzewa decyzyjnego z maksymalną głębokością 3. Test przeprowadzono dla różnych funkcji wyłaniających ekspertów. CLFE to podstawowa wersja klasyfikatora, eksperci klasowi wyłaniani są na podstawie najlepszej precyzji. W CLFE F1 eksperci wyłaniani są na podstawie miary F-1 klas. Natomiast w CLFE G wyłonienie ekspertów odbywa się na podstawie miary G-mean. Skrót CV, oznacza wersję ze sprawdzianem krzyżowym. Wyniki zostały porównane do drzewa decyzyjnego (TREE) z maksymalną głębokością równą 3. Dokładność klasyfikacji (tabela 5.1) oraz czułość klasy większościowej (tabela 5.2) wzrosła dla większości zbiorów danych. Dla pozostałych zbiorów, wyniki były takie same jak drzewa decyzyjnego lub minimalnie niższe. Natomiast specyficzność klasy mniejszościowej (tabela 5.3) oraz miara G-mean (tabela 5.4) wzrosły lub utrzymały taką samą wartości jak drzewa decyzyjne w 21 zbiorach na 22. W wykrywaniu klasy

mniejszościowej najlepsze okazały się klasyfikatory oparte o miarę F1 i G-mean. Pomiędzy wersją klasyfikatora ze sprawdzianem krzyżowym oraz bez, nie zauważono dużej różnicy w wynikach w większości zbiorów.

Zbiór danych	TREE	CLFE	CLFE CV	CLFE F1	CLFE F1 CV	CLFE G	CLFE G CV
seeds	0.91	**0.92**	0.91	0.91	0.91	0.91	0.91
new_thyroid	**0.97**	**0.97**	**0.97**	**0.97**	0.96	**0.97**	0.96
vehicle	0.9	0.9	0.9	**0.92**	**0.92**	**0.92**	0.9
ionosphere	0.86	0.88	**0.89**	0.86	0.87	0.86	0.87
vertebal	0.71	0.74	0.73	0.73	0.77	0.73	**0.78**
yeastME3	0.94	**0.95**	**0.95**	**0.95**	0.94	**0.95**	0.94
ecoli	0.86	0.85	0.85	0.87	0.87	**0.88**	0.76
bupa	0.65	**0.69**	0.68	0.68	0.67	0.68	0.6
horse_colic	**0.86**	**0.86**	0.84	**0.86**	**0.86**	**0.86**	**0.86**
german	0.74	**0.76**	0.75	0.73	**0.76**	0.73	0.74
breast_cancer	**0.73**	0.71	0.71	0.7	0.72	0.72	0.72
cmc	**0.78**	0.75	0.76	0.74	0.74	0.7	0.68
hepatitis	0.66	**0.68**	**0.68**	0.66	0.66	0.61	0.66
haberman	**0.75**	0.73	0.74	0.74	0.74	0.74	0.69
transfusion	0.68	**0.69**	0.68	0.68	**0.69**	0.68	**0.69**
car	0.69	0.89	**0.92**	**0.92**	**0.92**	0.89	0.89
glass	**0.82**	0.63	0.8	**0.82**	0.74	0.49	0.48
abalone16_29	**0.94**	**0.94**	**0.94**	0.93	0.88	0.68	0.68
solar_flare	**0.95**	0.81	0.94	**0.95**	0.87	0.65	0.65
heart_cleveland	**0.86**	**0.86**	0.83	0.85	0.83	0.81	0.81
balance_scale	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**
postoperative	0.68	0.69	0.71	0.67	**0.72**	0.67	0.67

Tablica 5.1: Dokładność klasyfikatora eksperckiego.

Zbiór danych	TREE	CLFE	CLFE CV	CLFE F1	CLFE F1 CV	CLFE G	CLFE G CV
seeds	**0.92**	**0.92**	0.91	**0.92**	**0.92**	**0.92**	**0.92**
new_thyroid	0.98	0.98	**0.99**	0.98	**0.99**	0.98	0.97
vehicle	0.89	0.89	0.89	**0.95**	**0.95**	**0.95**	0.89
ionosphere	0.92	0.97	**0.99**	0.92	0.93	0.92	0.93
vertebal	0.71	**0.72**	0.71	0.71	0.71	0.71	**0.72**
yeastME3	0.97	0.97	0.97	**0.98**	0.97	0.97	0.97
ecoli	0.9	0.88	0.86	**0.91**	0.89	0.9	0.77
bupa	0.72	0.77	0.81	0.81	0.72	**0.82**	0.63
horse_colic	**0.92**	**0.92**	0.88	**0.92**	**0.92**	**0.92**	**0.92**
german	0.88	**0.89**	**0.89**	0.86	0.84	0.77	0.81
breast_cancer	**0.92**	0.88	0.85	0.85	0.86	0.84	0.84
cmc	**0.9**	0.86	0.86	0.88	0.83	0.75	0.7
hepatitis	0.7	**0.73**	**0.73**	0.7	0.67	0.59	0.63
haberman	0.91	0.91	**0.93**	0.9	**0.93**	0.88	0.85
transfusion	0.76	**0.81**	0.8	0.8	**0.81**	0.76	0.8
car	0.71	0.89	**0.94**	**0.94**	**0.94**	0.89	0.89
glass	**0.88**	0.64	0.85	**0.88**	0.78	0.48	0.45
abalone16_29	**1.0**	0.99	0.99	0.99	0.92	0.69	0.69
solar_flare	**0.99**	0.83	0.98	0.98	0.88	0.64	0.64
heart_cleveland	**0.97**	**0.97**	0.89	0.91	0.89	0.83	0.83
balance_scale	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
postoperative	0.9	0.89	0.94	0.83	**0.95**	0.83	0.85

Tablica 5.2: Czułość klasy większościowej dla klasyfikatora eksperckiego.

Zbiór danych	TREE	CLFE	CLFE CV	CLFE F1	CLFE F1 CV	CLFE G	CLFE G CV
seeds	0.89	**0.91**	0.9	0.89	0.9	0.89	0.9
new_thyroid	**0.87**	**0.87**	0.8	**0.87**	0.8	**0.87**	**0.87**
vehicle	**0.93**	**0.93**	**0.93**	0.84	0.84	0.84	**0.93**
ionosphere	0.75	0.71	0.71	0.75	**0.76**	0.75	**0.76**
vertebal	0.71	0.76	0.76	0.76	**0.9**	0.76	**0.9**
yeastME3	0.73	**0.77**	**0.77**	0.71	0.7	0.75	0.73
ecoli	0.49	0.6	0.71	0.51	0.63	**0.74**	0.69
bupa	0.55	0.57	0.5	0.5	**0.61**	0.48	0.57
horse_colic	0.75	0.75	**0.78**	0.75	0.76	0.75	0.76
german	0.42	0.46	0.44	0.44	0.55	**0.62**	0.57
breast_cancer	0.31	0.31	0.39	0.35	0.41	0.42	**0.44**
cmc	0.39	0.37	0.42	0.28	0.45	0.51	**0.61**
hepatitis	0.5	0.5	0.5	0.5	0.62	0.69	**0.78**
haberman	0.32	0.23	0.21	0.28	0.2	**0.35**	0.25
transfusion	**0.45**	0.32	0.3	0.31	0.3	0.41	0.35
car	0.32	**1.0**	0.43	0.43	0.43	**1.0**	**1.0**
glass	0.12	0.47	0.24	0.12	0.29	0.65	**0.82**
abalone16_29	0.09	0.13	0.15	0.13	0.28	**0.58**	**0.58**
solar_flare	0.12	0.44	0.12	0.09	0.63	**0.93**	**0.93**
heart_cleveland	0.03	0.03	0.34	0.37	0.31	**0.63**	**0.63**
balance_scale	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
postoperative	0.08	0.12	0.08	**0.21**	0.08	**0.21**	0.17

Tablica 5.3: Specyficzność klasy mniejszościowej dla klasyfikatora eksperckiego.

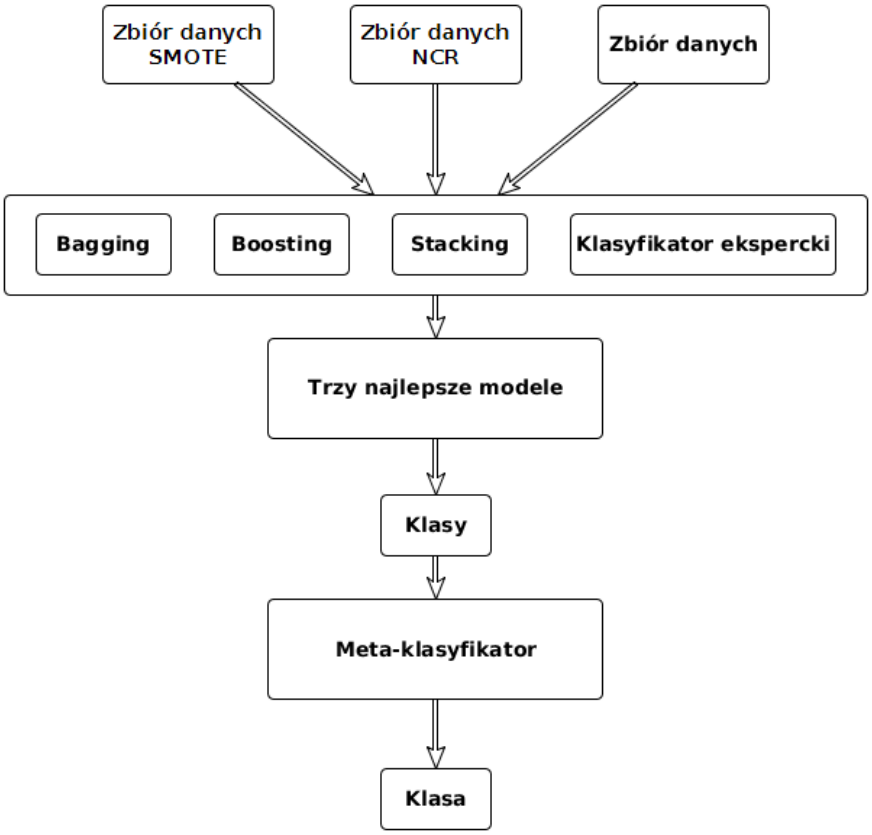
Zbiór danych	TREE	CLFE	CLFE CV	CLFE F1	CLFE F1 CV	CLFE G	CLFE G CV
seeds	0.9	**0.92**	0.91	0.9	0.91	0.9	0.91
new_thyroid	**0.92**	**0.92**	0.89	**0.92**	0.89	**0.92**	**0.92**
vehicle	**0.91**	**0.91**	**0.91**	0.89	0.89	0.89	**0.91**
ionosphere	0.83	0.83	**0.84**	0.83	**0.84**	0.83	**0.84**
vertebal	0.71	0.74	0.74	0.74	**0.8**	0.74	**0.8**
yeastME3	0.84	**0.86**	**0.86**	0.83	0.82	0.85	0.84
ecoli	0.66	0.73	0.79	0.69	0.75	**0.82**	0.73
bupa	0.63	**0.67**	0.64	0.63	0.66	0.63	0.6
horse_colic	**0.83**	**0.83**	**0.83**	**0.83**	**0.83**	**0.83**	**0.83**
german	0.61	0.64	0.63	0.61	0.68	**0.69**	0.68
breast_cancer	0.53	0.52	0.57	0.55	0.59	**0.6**	**0.6**
cmc	0.59	0.56	0.6	0.49	0.61	0.62	**0.65**
hepatitis	0.59	0.6	0.6	0.59	0.65	0.63	**0.7**
haberman	0.54	0.46	0.44	0.5	0.43	**0.55**	0.46
transfusion	**0.59**	0.51	0.49	0.5	0.49	0.56	0.53
car	0.48	**0.94**	0.64	0.63	0.63	**0.94**	**0.94**
glass	0.32	0.55	0.45	0.32	0.48	0.56	**0.61**
abalone16_29	0.3	0.35	0.39	0.35	0.5	**0.63**	**0.63**
solar_flare	0.34	0.61	0.34	0.3	0.74	**0.77**	**0.77**
heart_cleveland	0.17	0.17	0.55	0.58	0.53	**0.72**	**0.72**
balance_scale	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
postoperative	0.27	0.33	0.28	**0.42**	0.28	**0.42**	0.38

Tablica 5.4: Miara G-mean dla klasyfikatora eksperckiego.

5.2 Meta-klasyfikator

Algorytm klasyfikacji danych wybiera się w zależności od charakterystyki i rodzaju danych. Budowany meta-klasyfikator miał osiągać dobre wyniki dla różnych

zbiorów danych. Architektura zbudowanego meta-klasyfikatora jest bardzo złożona. Została przedstawiona na rysunku 5.2. W pierwszej warstwie składa się on z klasyfikatora eksperckiego oraz z kilku klasyfikatorów bagging, boosting oraz stacking. Tworząc meta-klasyfikator należy wybrać z jakich klasyfikatorów utworzone będą klasyfikatory bagging, boosting, stacking oraz klasyfikator ekspercki. Można wybrać kilka klasyfikatorów do utworzenia kilku klasyfikatorów bagging oraz boosting. Klasyfikatorami składowymi mogą być różne algorytmy klasyfikacji lub takie same algorytmy, ale z innymi ustawieniami. Klasyfikator ekspercki oraz stacking tworzone są z tych samych klasyfikatorów składowych.



Rysunek 5.2: Projekt meta-klasyfikatora.

W kolejnym kroku oceniany jest każdy model klasyfikacyjny poprzez sprawdzian krzyżowy dla niemodyfikowanego zbioru danych, dla zmodyfikowanego metodą NCR oraz dla zmodyfikowanego metodą SMOTE. Każda instancja klasyfikatora jest najpierw kopiowana, a następnie uczona na kolejnym zbiorze danych. Do końcowej klasyfikacji wybierane są trzy najlepsze klasyfikatory ze zbiorami danych, dla których osiągnęły najlepszy wynik. Kryterium wyboru najlepszych klasyfikatorów stanowi miara G-mean. W ostatnim etapie wybrane klasyfikatory trenowane są na pełnym zbiorze danych, a następnie klasyfikują ten zbiór danych, tworząc meta-dane stanowiące zbiór uczący dla końcowego meta-klasyfikatora - sieci neuronowej wielowarstwowej. Ostateczną klasę klasyfikowanemu przykładowi nadaje sieć neuronowa. Kod klasyfikatora znajduje się w pliku $classifiers/meta\_clf.py$.

5.2.1 Testy

Do stworzenia testowego meta-klasyfikatora wybrano klasyfikator kNN, drzewo decyzyjne z maksymalną głębokością 3 oraz naiwny klasyfikator Bayesa. Z tych trzech klasyfikatorów utworzono klasyfikator ekspercki, stacking oraz trzy modele klasyfikacyjne bagging. Utworzono również dwa modele AdaBoost z drzewa decyzyjnego z maksymalną głębokością 3 oraz z naiwnego klasyfikatora Bayesa (klasyfikator kNN nie wspiera wag). Każdy utworzony klasyfikator składowy został przetestowany dla zbioru danych bez modyfikacji, dla zbioru zmodyfikowanego metodą NCR oraz dla zbioru zmodyfikowanego metodą SMOTE. Dla takiej konstrukcji może wystąpić przypadek wyboru np. trzech modeli (kopii) tego samego klasyfikatora bagging, zbudowanych dla różnych zbiorów danych. Przeprowadzony test znajduje się w pliku *test_my_clfs/test_meta_clf.py*. Otrzymane wyniki porównano do meta-metod oraz klasyfikatora eksperckiego, czyli klasyfikatorów wchodzących w skład meta-klasyfikatora. W tabeli 5.5 przedstawiono dokładność, a w tabeli 5.6 czułość klasy większościowej meta-klasyfikatora. Nie udało uzyskać się najlepszego wyniku dla

Zbiór danych	Bag kNN	Bag TREE	Bag NKB	Ada TREE	Ada NKB	CLFE	META
seeds	**0.83**	0.79	0.8	0.79	0.79	0.79	0.8
new_thyroid	0.96	**0.97**	0.96	**0.97**	0.95	**0.97**	**0.97**
vehicle	0.93	0.91	0.67	**0.98**	0.85	0.93	0.97
ionosphere	0.77	**0.89**	0.86	0.86	**0.89**	0.84	0.85
vertebal	0.72	0.72	**0.78**	0.72	0.73	0.68	0.74
yeastME3	**0.95**	**0.95**	0.26	0.94	0.79	0.78	**0.95**
ecoli	0.84	0.81	0.84	**0.87**	0.43	0.8	0.78
bupa	**0.69**	0.65	0.55	0.67	0.58	0.6	0.65
horse_colic	0.72	**0.85**	0.77	0.82	0.72	0.77	0.8
german	0.7	**0.76**	0.72	0.72	0.62	0.68	0.74
breast_cancer	0.66	**0.73**	0.72	0.68	0.51	0.69	0.64
cmc	0.77	**0.79**	0.68	0.73	0.63	0.7	0.74
hepatitis	0.71	0.75	0.68	**0.81**	0.72	0.61	0.77
haberman	0.63	0.57	**0.72**	0.68	0.52	0.48	0.6
transfusion	0.72	**0.76**	0.74	0.64	0.59	0.65	0.58
car	0.97	0.95	0.89	**0.99**	0.96	0.89	0.97
glass	0.82	0.75	0.6	0.8	**0.87**	0.51	0.79
abalone16_29	**0.94**	**0.94**	0.68	0.92	0.61	0.91	0.9
solar_flare	**0.95**	**0.95**	0.67	0.93	0.58	0.94	0.94
heart_cleveland	**0.88**	**0.88**	0.82	0.86	0.78	0.85	0.82
balance_scale	**0.92**	**0.92**	**0.92**	0.87	**0.92**	0.81	0.85
postoperative	0.62	0.67	0.62	0.57	**0.7**	0.57	0.64

Tablica 5.5: Dokładność meta-klasyfikatora.

każdego zbioru, ale otrzymane wyniki zdecydowanie plasują się w czołówce. Jednocześnie udało się uniknąć i utrzymać wysoką jakość klasyfikacji dla wszystkich baz (np. klasyfikator bagging NKB, mimo wysokich wyników, dla *yeastME3* uzyskał tylko 26% dokładności). Jeżeli chodzi o rozpoznawalność klasy mniejszościowej (tabela specyficzności 5.7) oraz miarę G-mean (tabela 5.8), to otrzymane wyniki są

Zbiór danych	Bag kNN	Bag TREE	Bag NKB	Ada TREE	Ada NKB	CLFE	META
seeds	**0.79**	0.74	0.74	0.74	0.73	0.73	0.73
new_thyroid	**1.0**	0.98	0.98	0.98	0.97	0.98	0.99
vehicle	0.95	0.92	0.62	**0.98**	0.91	0.96	0.97
ionosphere	**0.99**	0.92	0.88	0.88	0.96	0.86	0.87
vertebal	0.71	0.72	**0.73**	0.71	0.67	0.63	0.72
yeastME3	**0.98**	0.97	0.17	0.97	0.87	0.78	0.97
ecoli	0.87	0.84	0.83	**0.91**	0.37	0.78	0.8
bupa	**0.81**	0.76	0.38	0.74	0.74	0.57	0.59
horse_colic	0.8	**0.91**	0.79	0.88	0.81	0.81	0.83
german	0.87	**0.92**	0.75	0.83	0.76	0.76	0.84
breast_cancer	0.88	**0.92**	0.84	0.75	0.41	0.8	0.7
cmc	0.91	**0.93**	0.7	0.84	0.74	0.79	0.85
hepatitis	**0.89**	0.8	0.65	0.87	0.81	0.59	0.87
haberman	0.65	0.54	**0.89**	0.77	0.45	0.39	0.62
transfusion	0.86	0.91	**0.92**	0.75	0.68	0.76	0.67
car	**0.99**	0.98	0.89	**0.99**	0.97	0.88	0.98
glass	0.88	0.81	0.59	0.85	**0.94**	0.48	0.85
abalone16_29	0.99	**1.0**	0.68	0.96	0.63	0.95	0.94
solar_flare	**0.99**	**0.99**	0.66	0.97	0.59	0.97	0.97
heart_cleveland	**1.0**	0.99	0.86	0.97	0.85	0.93	0.89
balance_scale	**1.0**	**1.0**	**1.0**	0.94	**1.0**	0.88	0.92
postoperative	0.82	0.88	0.8	0.71	**0.94**	0.68	0.83

Tablica 5.6: Czułość klasy większościowej meta-klasyfikatora.

powyżej średniej. Również tutaj udało uniknąć się bardzo niskich wyników, jakie zdarzały się pojedynczym klasyfikatorom, np. klasyfikator bagging kNN miał bardzo niską rozpoznawalność klasy mniejszościowej w bazie *german*, *breast cancer*, *hepatitis*, *glass*, *heart cleveland*. Natomiast meta-klasyfikator uzyskał wyniki plasujące go w czołówce dla wszystkich zbiorów danych.

Stworzony projekt meta-klasyfikatora stanowi interesującą propozycję uniwersalnego klasyfikatora. Skuteczność meta-klasyfikatora można poprawić poprzez zastosowanie bardziej różnorodnych algorytmów klasyfikacyjnych (np. można dodatkowo zastosować sieć neuronową) lub bardziej zaawansowanych metod równoważenia zbiorów danych.

Zbiór danych	Bag kNN	Bag TREE	Bag NKB	Ada TREE	Ada NKB	CLFE	META
seeds	0.93	0.87	0.93	0.89	0.91	0.9	**0.94**
new_thyroid	0.73	**0.87**	**0.87**	**0.87**	**0.87**	**0.87**	0.83
vehicle	0.87	0.89	0.84	**0.95**	0.64	0.84	**0.95**
ionosphere	0.4	**0.83**	**0.83**	**0.83**	0.76	0.8	**0.83**
vertebal	0.73	0.72	**0.88**	0.75	0.84	0.78	0.79
yeastME3	0.69	0.8	**0.99**	0.69	0.21	0.77	0.8
ecoli	0.57	0.6	0.94	0.51	**0.97**	0.94	0.6
bupa	0.52	0.49	**0.79**	0.57	0.35	0.63	0.73
horse_colic	0.59	**0.75**	0.74	0.73	0.58	0.71	**0.75**
german	0.3	0.37	**0.66**	0.46	0.3	0.49	0.5
breast_cancer	0.16	0.28	0.44	0.51	**0.74**	0.42	0.51
cmc	0.27	0.28	**0.61**	0.32	0.26	0.36	0.34
hepatitis	0.03	0.53	**0.78**	0.59	0.34	0.72	0.38
haberman	0.57	0.67	0.25	0.41	0.69	**0.73**	0.53
transfusion	0.25	0.27	0.18	0.28	0.29	**0.31**	**0.31**
car	0.58	0.11	**1.0**	**1.0**	0.78	**1.0**	0.65
glass	0.06	0.12	0.71	0.24	0.12	**0.82**	0.12
abalone16_29	0.11	0.07	**0.58**	0.28	0.31	0.3	0.4
solar_flare	0.0	0.02	**0.93**	0.12	0.28	0.14	0.21
heart_cleveland	0.0	0.06	**0.51**	0.06	0.23	0.31	0.26
balance_scale	0.0	0.0	0.0	**0.08**	0.0	**0.08**	0.06
postoperative	0.08	0.08	0.12	0.17	0.04	**0.25**	0.12

Tablica 5.7: Specyficzność klasy mniejszościowej meta-klasyfikatora.

Zbiór danych	Bag kNN	Bag TREE	Bag NKB	Ada TREE	Ada NKB	CLFE	META
seeds	**0.85**	0.8	0.83	0.81	0.82	0.81	0.83
new_thyroid	0.86	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**	0.91
vehicle	0.91	0.9	0.72	**0.97**	0.76	0.9	0.96
ionosphere	0.63	**0.87**	0.85	0.85	0.85	0.83	0.85
vertebal	0.72	0.72	**0.8**	0.73	0.75	0.7	0.75
yeastME3	0.83	**0.88**	0.42	0.82	0.42	0.77	**0.88**
ecoli	0.7	0.71	**0.88**	0.68	0.6	0.86	0.69
bupa	0.65	0.61	0.55	0.65	0.51	0.6	**0.66**
horse_colic	0.68	**0.82**	0.77	0.8	0.68	0.76	0.79
german	0.51	0.58	**0.7**	0.62	0.48	0.61	0.65
breast_cancer	0.38	0.51	0.6	**0.62**	0.55	0.58	0.59
cmc	0.5	0.51	**0.65**	0.52	0.43	0.53	0.54
hepatitis	0.17	0.65	0.71	**0.72**	0.53	0.65	0.57
haberman	**0.61**	0.6	0.47	0.56	0.56	0.53	0.57
transfusion	0.46	**0.5**	0.41	0.46	0.44	0.49	0.46
car	0.76	0.32	0.94	**1.0**	0.87	0.94	0.8
glass	0.23	0.31	**0.65**	0.45	0.33	0.63	0.32
abalone16_29	0.34	0.26	**0.63**	0.52	0.44	0.54	0.61
solar_flare	0.0	0.15	**0.78**	0.34	0.41	0.37	0.45
heart_cleveland	0.0	0.24	**0.67**	0.23	0.44	0.54	0.48
balance_scale	0.0	0.0	0.0	**0.28**	0.0	0.27	0.24
postoperative	0.26	0.27	0.32	0.34	0.2	**0.41**	0.32

Tablica 5.8: Miara G-mean meta-klasyfikatora.

Rozdział 6

Podsumowanie

W pracy przedstawiono algorytmy klasyfikacji oraz trzy meta-metody: bagging, boosting, stacking. Pokazano w jaki sposób należy oceniać klasyfikator. Przeprowadzono badania jakości klasyfikacji z użyciem meta-metod dla kilku różnych algorytmów klasyfikacji. Do testów użyto zróżnicowane (pod względem liczby przykładów, atrybutów, liczby atrybutów kategorycznych, różnym stopniu niezrównoważenia klas) i prawdziwe zbiory danych. Analiza otrzymanych wyników oraz porównanie ich do podstawowych klasyfikatorów wykazała, że z wykorzystaniem meta-metod można podnieść jakość klasyfikacji. Najlepszymi meta-klasyfikatorami okazały się bagging z naiwnym klasyfikatorem Bayesa oraz drzewem decyzyjnym. Algorytm AdaBoost z drzewem decyzyjnym uzyskał minimalnie gorsze wyniki. Stacking natomiast był najbardziej stabilnym meta-klasyfikatorem, uzyskując wysoką skuteczność klasyfikacji dla większości baz danych. Użycie klasyfikatora kNN, jako klasyfikatora składowego meta-metod, nie poprawiło jakości klasyfikacji.

Meta-metody z różnym skutkiem klasyfikowały klasę mniejszościową. Użycie metod równoważenia liczebności klas w zbiorach z meta-metodami pozwoliło zwiększyć wykrywalność klasy mniejszościowej o kilka procent, a w niektórych przypadkach kilkukrotnie. Najbardziej skutecznymi metodami okazały się metoda ADASYN oraz metoda NCR.

W ramach pracy zaprojektowano klasyfikator ekspercki oraz meta-klasyfikator. Przeprowadzono testy klasyfikatora eksperckiego, które wykazały lepszą lub taką samą dokładność klasyfikacji jak drzewo decyzyjne w większości zbiorów danych. W prawie wszystkich zbiorach poprawił lub utrzymał skuteczność klasyfikacji klasy mniejszościowej. Wyniki stworzonego meta-klasyfikatora były porównywalne z wynikami meta-metod. Kluczową cechą zbudowanego meta-klasyfikatora miała być uniwersalność. Niezależnie od danych miał uzyskiwać wysokie wyniki. Analiza wyników wykazała, że meta-klasyfikator dobrze łączy przewidywania klasyfikatorów składowych i uzyskuje wysoką skuteczność klasyfikacji. Uzyskane wyniki plasowały się w czołówce dla wszystkich zbiorów danych. Meta-klasyfikator uzyskał większą jakość klasyfikacji klasy mniejszościowej od większości meta-metod.

Mimo złożoności meta-metod, meta-klasyfikatora i konieczności budowania wie-

lokrotnie różnych modeli klasyfikacyjnych to czas potrzebny na stworzenie modelu klasyfikacyjnego jest niski (dla przetestowanych zbiorów danych). Klasyfikacja nowych przykładów odbywała się bardzo szybko.

Badania nad skutecznością meta-metod można kontynuować i przeprowadzić z użyciem innych algorytmów klasyfikacji. Zastosowanie bardziej różnorodnych algorytmów klasyfikacji lub większej ilości klasyfikatorów w metodzie stacking powinno pozwolić na zwiększenie jakości klasyfikacji. Interesującym zagadnieniem może być klasyfikacja niezrównoważonych zbiorów danych z wykorzystaniem zmodyfikowanych meta-metod, zawierającymi w sobie algorytmy równoważenia liczebności zbiorów. Kolejnym pomysłem na poprawienie skuteczności klasyfikacji, może być dobór algorytmu w zależności od charakterystyki danych.

Bibliografia

[1] Batista G., Prati R. C., Monard M. C., A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter, 2004.

[2] Bishop C. M., Neural Networks for Pattern Recognition. Claredon press. Oxford, 1995.

[3] Breiman L., Bagging Predictors, Machine Learning, 1996.

[4] Breiman L., Random Forests. Machine Learning, 2001.

[5] Chawla N. V., Bowyer K. W., Hall L. O., Kegelmeyer W. P., SMOTE: synthetic minority over-sampling technique, 2002.

[6] Freund Y., Schapire R. E., A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, 1996.

[7] Freund Y., Schapire R. E., Experiments with a New Boosting Algorithm. Proceedings of the 13th International Conference on Machine Learning, Bari, 1996.

[8] He H., Bai Y., Garcia E. A., Li S., ADASYN: Adaptive synthetic sampling approach for imbalanced learning, 2008.

[9] He H., Garcia E. A., Learning from imbalanced data. IEEE Transactions on Data and Knowledge Engineering, 2009.

[10] Hu, F., Liu X., Dai J., Yu H., A Novel Algorithm for Imbalance Data Classification Based on Neighborhood Hypergraph.

[11] Long P., Servedio R., Random Classification Noise Defeats All Convex Potential Boosters.

[12] Kubat M., Matwin S., Addresing the curse of imbalanced training sets: one-side selection.

[13] Laurikkala J., Improving identification of difficult small classes by balancing class distribution. Technical report, University of Tampere, 2001.

[14] Napierala K., Stefanowski J., Identification of Different Types of Minority Class Examples in Imbalanced Data. In: E. Corchado, V. Snasel, A.Abraham, M. Wozniak et al. (eds): Hybrid Artificial Intelligence Systems, Proc. 7th Int Conference HAIS 2012.

[15] Stefanowski J., Dealing with Data Difficulty Factors while Learning from Imbalanced Data.

[16] Tomek I., Two modifications of CNN. IEEE Transactions on Systems Man and Communications, 1976.

[17] Wilson D. L., Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Communications, 1972.

[18] Wolpert D., Stacked Generalization, Neural Networks, 1992.

[19] S. Raschka. Mlxtend (machine learning extensions), https://github.com/rasbt/mlxtend

[20] Hockham N., Machine learning with imbalanced data sets https://www.youtube.com/watch?v=X9MZtvvQDR4

[21] The UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/

[22] Python Software Foundation. https://www.python.org/

[23] scikit-learn Machine Learning in Python. http://scikit-learn.org/

[24] imbalanced-learn. http://contrib.scikit-learn.org/imbalanced-learn/

Spis rysunków

Spis tablic